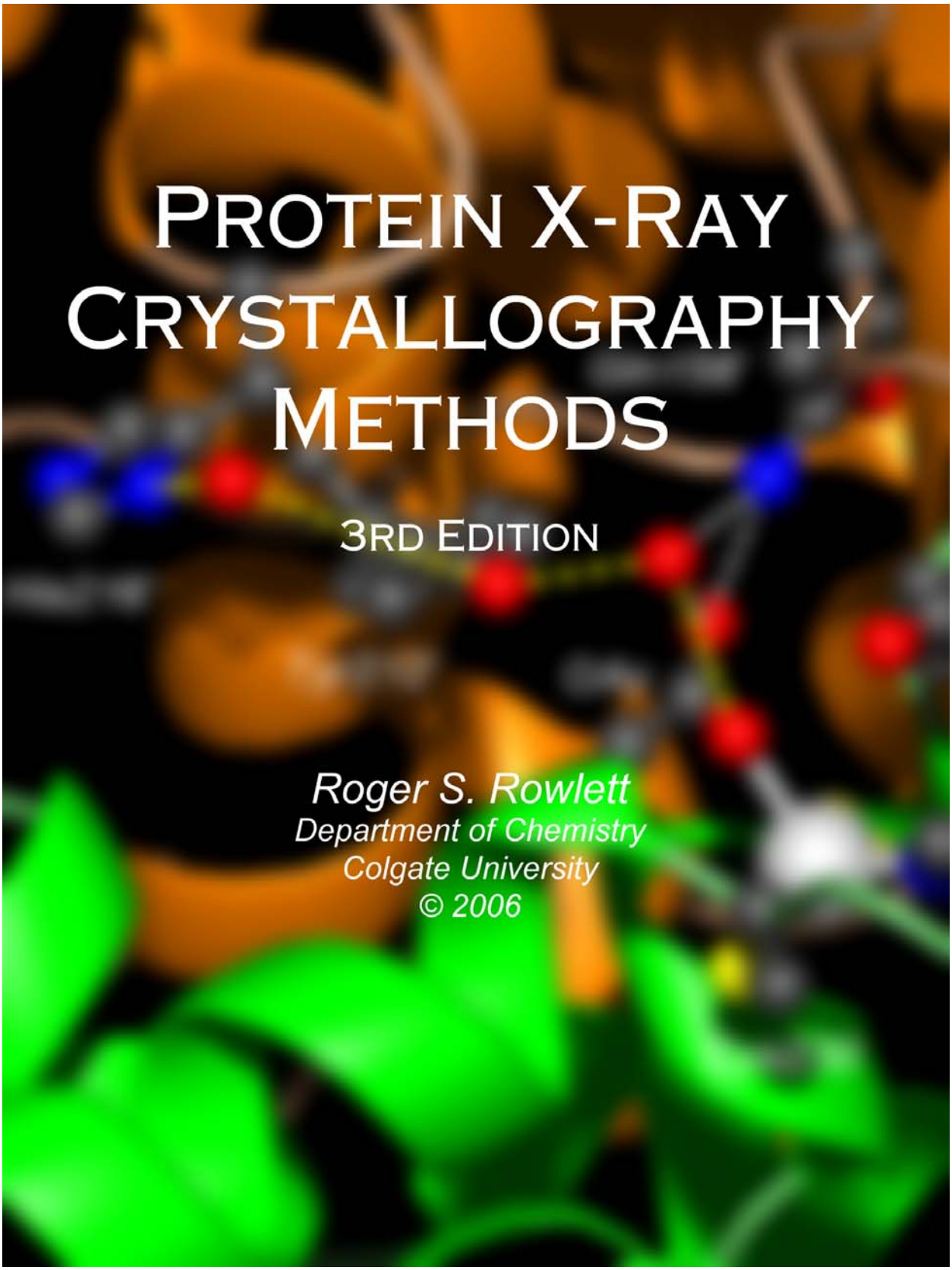

Protein X-Ray Crystallography Methods
3rd Edition
Roger S. Rowlett
Department of Chemistry
Colgate University
© 2006

Table of Contents

Introduction **1**

Abbreviations **1**

Crystallization of Proteins **3**
- Preparation of protein samples 3
- Hanging drop crystallization 3
- Initial screening of crystallization conditions 4
- Optimizing crystallization conditions 5
- Additional resources 6

Soaking, Mounting, and Freezing Protein Crystals **7**
- Screening for a suitable cryoprotectant 7
- Mounting protein crystals on loops 8
- Soaking-in ligands 9
- Additional resources 9

X-Ray Diffraction Data Collection **11**
- Preparing the diffractometer for data collection 11
- Screening crystals for diffraction 11
- Collecting a complete diffraction data set 14
- Shutdown procedure 14

Bare-bones Linux 17
- Starting and ending a Linux session 18
- File storage Structure in Linux 18
- Commonly used Linux commands 18
- Special Linux command line characters and actions 20
- Input/output redirection 21
- Customizing your Linux environment 21
- Additional resources 22

X-ray Diffraction Data Analysis **23**
- Indexing the first frame using DENZO 23
- Integrating an entire data set using DENZO 26
- Scaling reflection data set using SCALEPACK 28
- Indexing the first frame using MOSFLM 30
- Integrating the entire data set using MOSFLM 32
- Scaling reflection data using SCALA 34
- Merging multiple data sets in CCP4i 36
- Reindexing data sets in CCP4i 38
- Additional resources 39

Model Building and Refinement.....41
Preparing SCALEPACK data for analysis 41
Converting MTZ files to CNS format.....42
Obtaining phases by molecular replacement 43
Constructing a molecular replacement model 44
Preparing the first electron density map 50
Finding molecular replacement solutions using EPMR 44
Finding molecular replacement solutions using Phaser.....47
Initial model refinement using CNS 50
Further model refinement using CNS 52
Refining structures using CCP4.....54
Visualizing molecules and electron density maps in O 48
Basic model building tasks in O 62
Adding ligands and cofactors to a model in O.....64
Adding water molecules to a model using CNS and O 65
Visualizing molecules and electron density maps in Coot 65
Basic model building tasks in Coot 68
Adding ligands and cofactors using Coot 69
Adding water molecules to a model using CCP4 and Coot.....69
Validating structures 71
Additional resources 71

Introduction

This first edition of this manual was originally written in May 2003 to provide a compendium of up-to-date commonly used methods routinely used in the X-ray crystallography laboratory in the Molecular Structure Section of the Laboratory of Molecular Biology, National Institute of Diabetes and Digestive and Kidney Diseases at the National Institutes of Health. Much of the information contained in this manual was gleaned from my knowledgeable, kind, and very clever colleagues in the Laboratory of Dr. David Davies. Special thanks go to Drs. Thang Chiu and Jessica Bell, my very patient tutors. I am forever in debt to their mentorship.

The second edition was completed in May 2005 and incorporated additional material related to the conduct of protein X-ray crystallography as carried out in my undergraduate research laboratory at Colgate. I am indebted to my first "crystallography" research students, Ariel Herman ('04) and Joey Lee ('04) for their patience with the first edition. Their laboratory experience inspired improvements and updates to the second edition.

The third edition was completed in June 2006 and incorporates entirely new sections on using CCP4i, Refmac, Coot, and Phaser. With Coot, the CCP4 suite has become more accessible and easy-to-use than ever for undergraduates. My colleague Gino Cingolani deserves credit for convincing me to try Coot and Refmac in the undergraduate research environment. Many thanks—we like it! And my colleague Toshi Ohsumi (a con-conspirator in developing an open-source, parallelizable, genetic algorithm for molecular replacement) gets the nod for convincing me to try Phaser. Some cover art was added to make the manual look less cheesy and more recognizable in the lab.

Abbreviations Used

Some of the more common acronyms and abbreviations used in X-ray crystallography used in this manual are listed here.

- SDS-PAGE—sodium dodecyl sulfate polyacrylamide gel electrophoresis
- PEG—polyethylene glycol (typically followed by an average polymer molecular weight, *e.g.*, PEG-400 has an average molecular weight near 400)
- MPD— 2-Methyl-2,4-pentanediol

Crystallization of Proteins

Preparation Protein Samples

Purification. Protein samples should be as pure as possible for successful crystallization. Protein that is >90% pure should be sufficient for commencing crystallization screens. The more homogeneous the protein, the more likely crystallization is to be successful. Purity can be evaluated by SDS-PAGE, isoelectric focusing, and/or mass spectroscopy.

Sample storage. Proteins are typically stored at 4°C or frozen at –80°C in a solution appropriate to maintain stability and activity. Proteins solutions should be as concentrated as practical to enhance stability. A stock protein concentration of 10-20 mg/mL is typical for crystallization screening. If protein is stored frozen, it should be aliquoted to minimize repeated freeze/thaw cycles that are usually deleterious to proteins. In general, protein solutions should contain the minimum concentrations of buffers, salts, and preservatives necessary for safe storage. In particular, the use of high concentrations of glycerol or other polyols in storage solutions should be avoided, as this can alter or interfere with crystallization.

Sample handling. Most proteins are sensitive to harsh handling. Unless known otherwise, proteins should always be maintained on ice when not in the refrigerator, cold room, or freezer. In addition, protein solutions should *never* be subjected to vortexing or vigorous mixing; the resulting foaming promotes protein denaturation. Before using proteins in crystallization trials, it is customary to remove dust and precipitated protein by centrifugation at 14000 ×*g* for 5-10 minutes at 4 °C.

Hanging Drop Crystallization

The most common method of protein crystallization is *hanging drop vapor diffusion*. In this method, a concentrated protein solution is combined with a solution of a precipitant and allowed to concentrate by evaporation. Under the right conditions, and with the appropriate precipitant, protein crystals will form. In hanging drop vapor diffusion, a small volume of protein sample and precipitant are combined on a glass coverslip and sealed over a well containing precipitant solution (fig 1). Because the precipitant concentration in the mixed drop of protein is lower than in the well solution, water evaporates from the drop—increasing the concentration of both protein and precipitant—until the drop is in equilibrium with the well solution. The concentration of protein and precipitant in the drop occurs slowly and gradually, favoring crystallization over precipitation.

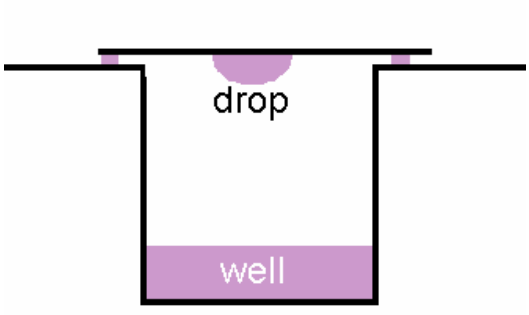


Figure 1. Hanging drop vapor diffusion

Preparing crystallization trays. Crystallization trials are conveniently performed in 24-well, pre-greased crystallization trays (fig 2). Prior to setting trays, carefully organize your solutions and record in your notebook the crystallization conditions to be used in each well. The following protocol is typical:

- Obtain a pre-greased 24-well crystallization tray and a box of 22 mm siliconized cover slips. If setting trays at 4 °C, allow the tray and all solutions to equilibrate before proceeding.
- Fill the wells of the tray with the appropriate precipitant solutions.
- Remove a coverslip from the box, taking care to handle only by the edges.
- Pipet 1 μL[1] of protein[2] on to the cover slip, taking care not to introduce bubbles.
- Pipet an equal volume of precipitant solution (from the corresponding well) into the protein drop and gently mix by pipetting up and down a few times.
- Immediately place the cover slip over the well, press down gently and twist 45° to ensure a good seal.
- Repeat for remaining wells.
- Immediately after preparing the plate, place it under a microscope and examine each of the drops for protein precipitation or foreign objects (glass shards, fibers, plastic bits) and make a notation of any drops that are not clear.
- Place the plate in a quiet place at the appropriate temperature and *leave it undisturbed for at least 24 hours*.

Figure 2. A Hampton Research 24-well VDX plate™ and siliconized coverslips

Initial Screening of Crystallization Conditions

Strategy. The determination of promising protein crystallization conditions is typically done using a sparse matrix screen, in which a protein is subjected to widely varying pH, salts and

[1] Up to 40 μL of protein can be used if desired.

[2] Protein concentrations from 5-20 mg/mL are typical; a protein concentration of ≈10 mg/mL is a good starting point for initial screens.

precipitants. There are excellent commercial screening kits available, making it generally unnecessary to mix your own initial screening reagents. The following commercial screens are recommended, in the order that they should be employed:

1. Hampton Research Crystal Screen. This screen contains 50 reagents. Screen conditions #25 and #27 have a historically poor record of producing crystals, and these two can be omitted in order to conduct the screen in two 24-well plates.
2. Hampton Research Crystal Screen 2. An extension of the original Hampton Research sparse matrix screen. Two conditions can be omitted for conducting a two-plate screen.

Evaluating screens. Plates should be examined under the microscope and evaluated for protein crystallization after 24 hours, and every day thereafter for a week. After one week, plates should be examined weekly. Record in your notebook results for each drop. Suggested categories and abbreviations to use are as follows, with comments:

- *Clear drop (C)*—no changes in drop
- *Precipitate (P)*—typically light brown and granular. Crystals sometimes form from such precipitates. If the precipitate is thick and swirly, this is very bad, and unlikely to form crystals.
- *Precipitate/phase separation (PP)*—typically light brown and granular, with little blobs that look like oil drops. Crystals sometimes from at the edge of phase separations.
- *Microcrystals (MX)*—Difficult to distinguish from precipitate; however, unlike precipitated protein, microcrystals have a shiny appearance. Optimization of conditions may result in larger crystals.
- *Needle cluster (NX)*—Beautiful, but often useless for crystallography. Optimization of conditions may result in less needle-like forms.
- *Plates (PX)*—if not too thin, may be useful for crystallography. Optimization may result in better crystal form. Plate clusters may be separated in to suitable single crystals.
- *Rod clusters (RX)*—If separable into single crystals, may be useful for crystallography
- *Single crystals (X)*—the Holy Grail: large, individual crystals with blocky dimensions.

Optimizing crystallization conditions

If more than half the drops are clear, you should consider increasing the protein concentration and re-screening. If most of the drops have copious precipitate, you should consider lowering the protein concentration and rescreening. *To save protein in the initial phase of screening, it may be advisable to run only half of a screen at a time in a 24-well plate until you establish the appropriate protein concentration for efficient screening.*

Conditions from the initial screen that show the most promise for crystallization should be further optimized in order to improve crystal form and size. To find the best crystallization conditions, pH, precipitant concentration, and protein concentration should be systematically varied. This will require stock solutions of concentrates so that a variety of custom solutions can be constructed for optimization. For example, typical buffer stock solutions are 1M and pre-adjusted to the desired pH. Salt solutions are generally prepared to near saturation, 1-4 M depending on the salt.

Crystal form can usually be further improved by exploring additives. Preformulated additive screens can be purchased (Hampton Research) or selected additives stocks at 10× final desired concentration can be prepared and added at 10% volume to hanging drops.

Additional resources

- Bergfors, Terese M. (1999) *Protein Crystallization, Techniques, Strategies, and Tips*, International University Line, La Jolla, CA.
- Hampton Research: http://www.hamptonresearch.com
- A practical guide to protein crystallization (Mark Knapp): http://www-structure.llnl.gov/crystal_lab/crystall.htm
- The Protein Crystallization Page (Terese Bergfors): http://xray.bmc.uu.se/~terese/
- X-tal Protocols (Johan Zeelan): http://www.mpibp-frankfurt.mpg.de/~johan.zeelen//xtal.html
- How to grow protein crystals: http://www.ccp14.ac.uk/ccp/web-mirrors/llnlrupp/crystal_lab/cystalmake.html
- Protein crystallography course: http://www-structmed.cimr.cam.ac.uk/Course/Crystals/intro.html

Soaking, Mounting and Freezing Protein Crystals

Most X-ray crystallographic data collection is done at low temperature (typically 100 K) to minimize degradation of the crystal by free radicals generated by the X-ray beam. This is especially important when using intense synchrotron X-ray sources. In order to prevent crystals from cracking when frozen, it is necessary to treat protein crystals with a cryoprotectant prior to freezing. In the presence of a cryoprotectant, the protein and its thin layer of surrounding mother liquor will form an amorphous glass in which the crystal suffers minimal damage, and retains maximum X-ray diffraction properties.

Screening for a suitable cryoprotectant

Unless the optimum crystallization conditions already contain a sufficient quantity of cryoprotectant, it will be necessary to experimentally determine solution conditions suitable for safely freezing crystals. Typically, some quantity of cryoprotectant is added to a solution of artificial mother liquor, or a solution of artificial mother liquor containing the appropriate amount of cryoprotectant is made up from scratch. Some typical cryoprotectants and concentrations required to assure proper freezing protection in the worst-case scenarios is given in Table 1 below. In many cases a lower concentration of cryoprotectant that that listed in Table 1 is sufficient. (For example, crystallization solutions already containing high concentrations of PEG may require little or no additional cryoprotection.) The minimum amount of cryoprotectant required can be determined by pipetting 10 μL drops of solution into liquid nitrogen. If drops reliably freeze clear, then the solution has sufficient cryoprotection for freezing protein crystals.

The choice of cryoprotectant will depend upon the crystallization solution composition. If protein crystallization conditions already contain a cryoprotectant, it is often ideal to simply increase the concentration to the appropriate value. This is especially convenient for PEG-containing solutions. However, PEGs have limited solubility in solutions that contain high concentrations of salt; in this case one of the other cryoprotectants in Table 1 is more likely to be suitable. Glycerol, glucose, or sucrose are very gentle to most proteins, have high solubility in a large variety of solution, and are often excellent choices.

Table 1
Typical Cryoprotecants and Concentrations Required

Cryoprotectant	Concentration
glycerol	30% v/v
sucrose	30% w/v
glucose	30% w/v
ethylene glycol	30% w/v
MPD	30% v/v
PEG 400-20000	25-40% (v/v or w/v)

Once a suitable cryoprotectant solution or solutions have been identified, the behavior of protein crystals in these solutions should be observed. This is often carried out at the same time as crystal mounting, as described below. You should observe that the crystal does not disintegrate, crack, or split during cryo-soaking. For especially difficult cases, you can try sequentially soaking crystals in 15% glucose and then 30% glucose prior to freezing. Many

proteins otherwise impossible to freeze survive this treatment. It is not necessary to soak crystals for extended periods to confer cryoprotection. All that is necessary is to replace the solution on the surface of the crystal with the cryoprotectant solution, a process that only takes a few seconds of soaking.

Mounting protein crystals on loops

Protein crystals are mounted for diffraction on tiny nylon loops 0.05–1.0 mm in diameter. The loops are mounted on hollow rods that are in turn mounted on magnetic caps that are conveniently stored under liquid nitrogen, and are easily placed on the goniometer head of the X-ray diffractometer. A photo of a loop and cap is shown below (fig 3). The following protocol is typical:

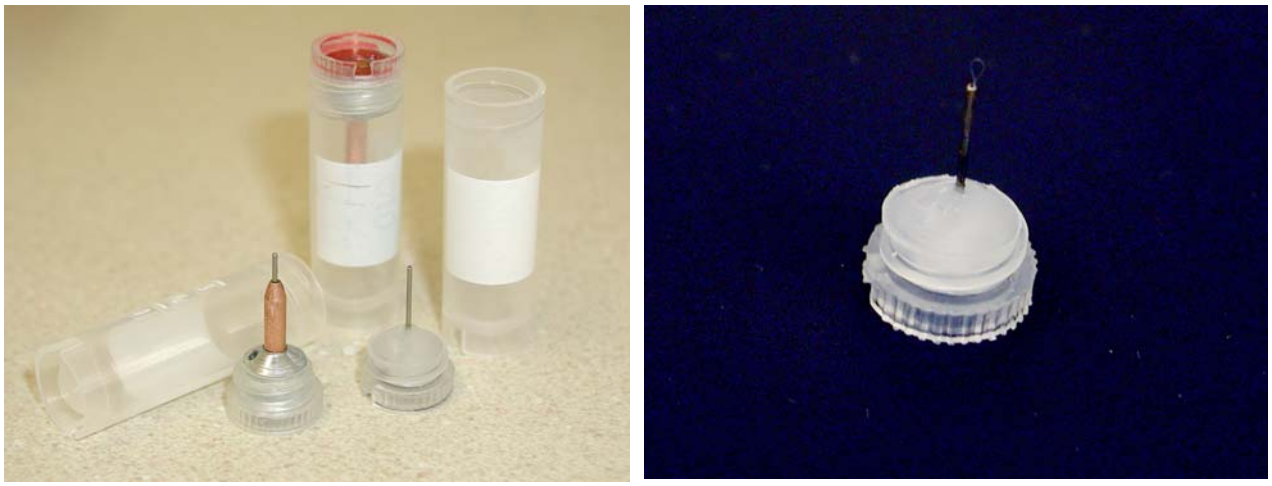


Figure 3. Mounting loops and cryovials (left). Closeup of a 0.50 mm mounting loop (right).

- Obtain and don comfortable Thinsulate gloves to protect your hands from frostbite.
- Fill a tall dewar with liquid nitrogen, and insert and cool a labeled cryo-cane to hold your mounted crystal samples. Fill a second dewar with liquid nitrogen to periodically top off the first dewar.
- Obtain a vial clamp and a crystal wand for handling vials and crystal caps.
- Obtain a collection of cryovials fitted with crystal caps with various sizes of mounting loops. The caps are color coded to aid in indentification.
- Obtain your crystal tray containing crystals to soak, freeze, and mount.
- Obtain a spot plate, a 20 uL pipettor and pipette tipes, and your cryoprotectant solutions.
- Assemble all of these materials around the dissecting microscope.
- Place the crystal tray under the microscope and focus on a well containing suitable crystals. Without removing the coverslip, determine what size loops are appropriate by holding them under the microscope next to the coverslip. You should choose a loop size that is just slightly larger than the crystals.
- Pipette 10-20 μL of cryoprotectant in a spot plate well

- Label[3] a cryovial bottom and mount it in the vial clamp
- *For the next steps you must work quickly, as the protein drop may evaporate rapidly, causing protein precipitation or crystal cracking.*[4]
- Carefully remove the coverslip with the desired crystals and place it drop-side up over an empty well of the spot plate.
- Mount an appropriate size loop on the crystal wand and fish out a crystal. The loop should be just larger than the crystal. If you maneuver the crystal close to the edge of the drop it will be easier to pick up.
- Place the crystal into the cryoprotectant solution by touching the loop to the drop.
- Observe the crystal under the microscope to check for cracking or disintegration. It is not necessary to soak the crystal for more than a few seconds in order to confer cryoprotection. If there are no problems, fish out the crystal in the loop and immediately plunge it into liquid nitrogen *and keep it there*. If the crystals crack or disintegrate, you need to find another cryo-soak.
- Immerse the empty cryovial into the liquid nitrogen until it stops bubbling. Keep both the crystal cap and the vial under the surface, and screw the crystal cap into the cryovial and mount the vial in the cane.
- Mount additional crystals as required before the drop evaporates or you run out of crystals.
- Store frozen cryovials in a liquid nitrogen storage dewar for future use.

Soaking-in ligands

Occasionally, it is desirable to determine a protein structure in the presence of a bound small molecule. One method of preparing such protein-ligand complexes is to soak a crystal in artificial mother liquor containing an excess of ligand; this can be done at the same time as cryoprotection if desirable and practical. Typically, the concentration of ligand used should be 10-1000× the dissociation constant (K_d) if it is known. Soaking for 10-30 min should be sufficient to populate the protein in the crystal if the binding site is accessible in the crystal lattice. If protein molecules pack in the crystal in such a way as to obscure the ligand-binding site, or if crystals do not tolerate extended soaking without cracking or dissolving, then co-crystallization with ligand should be attempted.

Additional resources

- Hampton Research: http://www.hamptonresearch.com
- Flash-Cooling: A Practical Guide: http://www.rose.brandeis.edu/PRLab/Crystalizations/cool/

[3] Each crystal should be labeled with a unique identifier so that it can be specifically identified later for diffraction screening and data collection. For example, Human carbonic anhydrase II crystals might be labeled HCAII-01, HCAII-02, *etc*. Cryocanes can be labeled with the first of a sequence of vial names contained within them for easy location in the storage dewar.

[4] Drop evaporation will be especially problematic during the winter months, when indoor humidity levels are very low. Working at 4 °C may minimize this problem.

X-ray Diffraction Data Collection

Before a data set can be collected—from which the final structure of the protein can be deduced—it is necessary to mount a crystal in the X-ray beam of a diffractometer and determine if it diffracts to sufficient resolution to justify collecting a full data set. The initial crystal screen can often be used to determine the space group of the crystal, an important piece of information is planning data collection. The exact instructions for collecting data will vary depending on the type of equipment used. The following instructions are suitable for using the Unix-controlled RAXIS-IV detector systems at the National Institutes of Health (Figure 4).

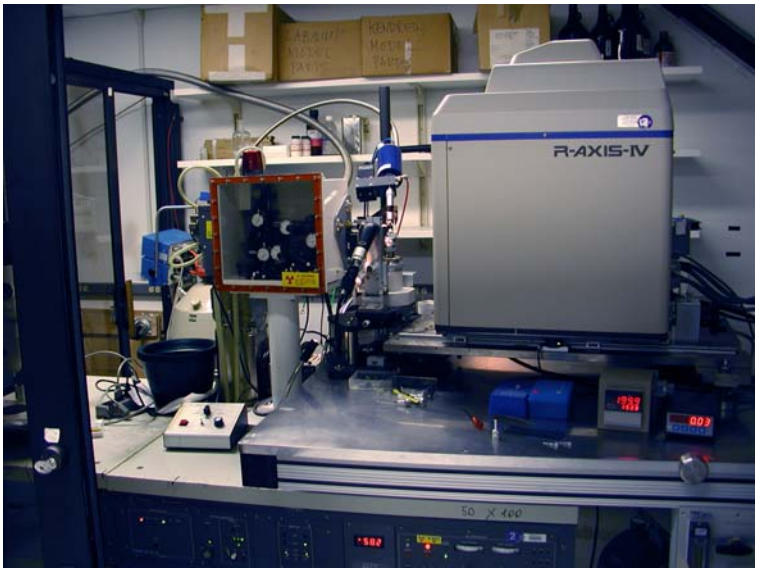


Figure 4. Source, cryosystream and RAXIS-IV detector system "B" at the National Institutes of Health, Laboratory of Molecular Biology, Bethesda, MD.

Screening crystals for diffraction

Preparing the diffractometer. The following steps must be carried out prior to commencing data collection:

- Configure and start the cryostream system to cool and hold your crystals to 95K. Depending on the particular cryo-system used, it can take up to 2 hours for the cryostream to come to temperature.
- Enter the X-ray hutch—*make sure that the X-ray beam is not on and the shutter is not open*!—and immediately verify that the X-ray shutter is manually *closed*. Closing the shutter when entering the hutch should become second nature. Closing the shutter minimizes the risk of being exposed to direct or backscattered X-ray radiation while working in the hutch.
- Immediately before collecting data, energize the X-ray source by (1) increasing the voltage to 50 kV and (2) increasing the current to 100 mA. These steps should be carried out in this order, and it is precautionary to increase both settings gradually as you bring up the source.

Collecting diffraction screen data. The following steps are typical for collecting crystal screens:

- Obtain and don comfortable Thinsulate gloves to protect your hands from frostbite.
- Obtain two tall dewars and fill them with liquid nitrogen. Extract the cryocane(s) with the appropriate crystals and place in one of the dewars.
- Obtain an empty crystal cap and loop the same size and type as those you will be using for data collection.
- Obtain a vial clamp, crystal wand, and cryo-tongs for handling your crystals, and take all the abovementioned items to the X-ray hutch.
- Move the cryostream head back to allow sufficient clearance for mounting crystals on the goniometer without touching the X-ray source, beam stop, or cryostream head.
- Mount the empty crystal cap and loop on the goniometer head and adjust the height so that the loop is centered in the X-ray beam (and cryostream). *Failure to do this prior to mounting your crystals may result in their melting before they can be centered in the cryostream.* Remove the empty cap and loop when adjustments have been completed.
- Using the vial clamp, remove the desired vial from the cryo-cane and keep it under the surface of the liquid nitrogen.
- Immerse the crystal wand into the liquid nitrogen until it stops bubbling vigorously. Use the crystal wand to unscrew the crystal cap with the mounted crystal. *Keep the crystal cap under the surface of the liquid nitrogen at all times.*
- Remove the empty vial and vial clamp from the liquid nitrogen and set aside. *Keep the crystal cap totally submerged in liquid nitrogern during this and the next two steps.*
- Insert the cryo-tongs into the liquid nitrogen and hold them there until all bubbling stops. This may take 30-60 seconds.The tongs will cool faster if they are held open during cooling. *It is important that the cryo-tongs are fully cooled before proceeding.*
- While keeping both tongs and crystal cap under the surface of the liquid nitrogen, open the tongs, grasp the crystal cap, and remove it from the crystal wand. The wand can now be removed from the liquid nitrogen and set aside.
- *Quickly* remove the cryo-tongs and the attached crystal cap from the liquid nitrogen and place the cap on the magnetic head of the goniometer. The tongs should be oriented so that when they are opened to release the cap, the crysostream can blow into the opening between the two halves of the tongs. This will prevent ice ring formation or crystal melting.
- Immediately adjust the goniometer head so that the crystal is centered vertically in the beam and properly centered in the cryostream.
- Unlock the ϕ-axis of the goniometer and swing it to 0°. Adjust the goniometer with the hex key until it is centered in the beam. Swing the goniometer to 90° and repeat. Do the same for 180° and 270° orientations. Repeat as necessary until the crystal is centered in the beam and rotates without wobbling in the center of the X-ray beam. You can check the accuracy of your alignment by looking for lateral displacement when the goniometer head is swung 180°: i.e., you should check that the crystal is not laterally displaced when swung 0-180°C or 90-270°. A properly mounted crystal is shown in Figure 5.
- When alignment is complete, swing the goniometer to 0° and lock the goniometer head.
- Remove all your tools and dewars from the hutch if you will need them during data collection, otherwise they can stay inside.

- Switch the X-ray shutter from CLOSED to EXTERNAL, exit the hutch, and close the door.
- Start the R-AXIS data collection software by entering the Unix command `start &` at the prompt.
- Enter your data collection parameters. For screening crystals it is useful to shoot three frames at 0°, 45° and 90° rotation about the ϕ-axis to evaluate diffraction along different crystal axes. If a practical camera distance is not known, start with 200 mm. This distance is sufficient to collect data to 2.5 Å at the edge of the frame. The oscillation range should be set to 0.2-2.0° or based on prior experience. Exposure times per frame of 2-20 minutes are typical.
- Initiate data collection.
- Individual frames should be analyzed by DENZO, MOSFLM, or other appropriate software for indexing and preliminary assignment of space group as described in the section on X-Ray Diffraction Data Analysis
- Perform the appropriate shutdown procedures (*vide infra*)

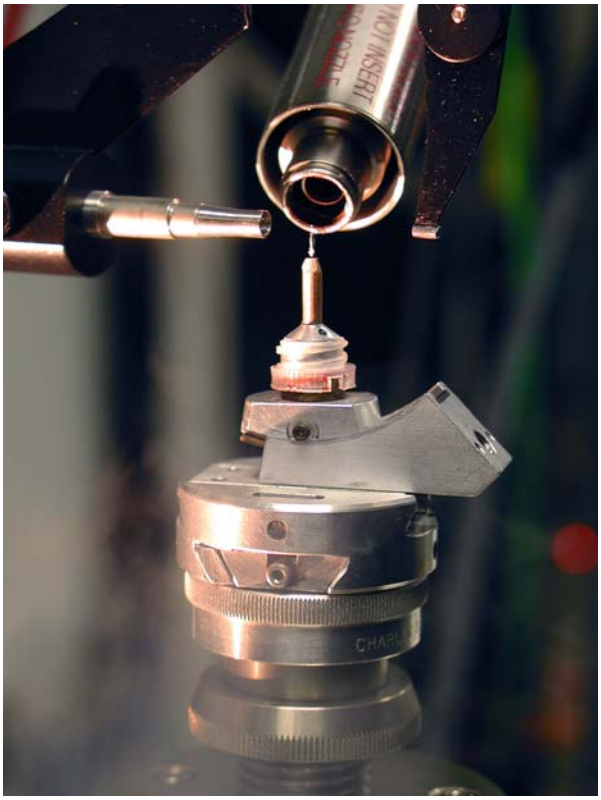


Figure 5. A crystal properly mounted and centered on the goniometer head and cooled by the cryoststream

Collecting a Complete Diffraction Data Set

The physical steps for collecting full data sets are nearly identical to that of collecting screen data:

- If your crystal is not already mounted and aligned, follow the instructions for mounting crystals as described in the previous section.
- Swing the goniometer to 0° or other desired starting angle and lock the goniometer head.
- Remove all your tools and dewars from the hutch. You will not be able to retrieve them during data collection.
- Switch the X-ray shutter from CLOSED to EXTERNAL, exit the hutch, and close the door.
- Start the R-AXIS data collection software (if it is not running already) by entering the Unix command `start &` at the prompt.
- Enter your data collection parameters. For data collection is is typical to collect data over a total ϕ rotation range of 45-180°. The camera distance should be set close enough to measure the highest resolution spots observable, but far enough away to resolve the closest spots in the diffraction patterns. For maximum efficienty in data collection, the oscillation range should be set to the maximum value tolerable before excessive spot overlap occurs. Exposure times should be long enough to accurately measure the intensities of the highest resolution spots, commensurate with the length of data collection time you have.
- Initiate data collection.
- After a 100 or more frames have been collected, they can be integrated and scaled by DENZO and SCALEPACK to assess the quality and completeness of data, as described in the section on Diffraction Data Analysis. When data collection is sufficiently complete, you can stop data collection and shut down the instrument as described below.

Shutdown Procedure

The X-ray diffractometer should be shut down in an orderly fashion in order to maximize the life of the X-ray source and prevent problems with the cryo-system. The following procedure should be employed:

- Obtain and don comfortable Thinsulate gloves to protect your hands from frostbite.
- Obtain two tall dewars and fill them with liquid nitrogen. Place the appropriate cryocane(s) (to store your retrieved crystal) in one of the dewars.
- Obtain a vial clamp, crystal wand, and cryo-tongs for handling your crystals, and take all the abovementioned items to the X-ray collection area.
- Ensure that data collection has stopped. If necessary use stop data collection manually using the emergency stop button in the data collection software. *Do not enter the hutch until you have verified that the X-ray beam shutter has been closed.*
- Enter the hutch and immediately switch the shutter from EXTERNAL to CLOSED.
- Return the X-ray source to minimum power by (1) turning the current down to the minimum level, and then (1) turning the voltage down to the minimum level.

- Move the cryostream head back to allow sufficient clearance for removing crystals from the goniometer without touching the X-ray source, beam stop, or cryostream head.
- Unlock the ϕ-axis of the goniometer and swing it to an angle where the notch on the crystal cap will line up with the notch on the cryotongs when they are used to remove to crystal.
- Clamp the empty cryovial in the vial clamp and set it aside.
- Insert the cryo-tongs into the liquid nitrogen and hold them there until all bubbling stops. This may take 30-60 seconds.The tongs will cool faster if they are held open during cooling. *It is important that the cryo-tongs are fully cooled before proceeding.*
- *Quickly* remove the cryo-tongs and remove the crystal cap from the goniometer. The tongs should be oriented so that when they are opened to grab the cap, the crysostream can blow into the opening between the two halves of the tongs. This will prevent ice ring formation or crystal melting.
- Immediately plunge the cryotongs and attached crystal cap into liquid nitrogen, and *leave it submerged.*
- Immerse the crystal wand into the liquid nitrogen until it stops bubbling vigorously. Use the crystal wand to remove the crystal cap from the cryotongs. *Keep the crystal cap under the surface of the liquid nitrogen at all times.*
- Plunge the vial clamp and attached vial into the liquid nitrogen and hold it there until it stops bubbling.
- While keeping both vial and crystal cap submerged in liquid nitrogen, use the crystal want to screw the crystal cap on to the vial. Set the crystal wand aside.
- Place the vial containing the crystal into the appropriate cryocane and return it to the tall dewar. The cryocane should be returned to its storage dewar when convenient.
- Power down the cryostream controller and attach the house nitrogen line to the cryostream head. This will prevent condensation and ice formation in the cryostream head.
- Remove your tools and supplies from the hutch.

Bare-Bones Linux

Because so many X-ray crystallography data analysis and protein modeling and refinement programs are written for the Unix/Linux platform, some familiarity with the Linux operating system is necessary to do protein crystallography. The following information is not intended to be an exhaustive Linux tutorial; rather, it is intended to be the bare minimum information required for starting, ending and navigating a Linux session using a workstation in the Colgate University Department of Chemistry Protein X-ray Crystallography Computing Facility. The latest information concerning the status of the Computing facility is available at http://departments.colgate.edu/chemistry/xray. Some helpful tricks and tips will be given along the way. You are encouraged to consult the additional resources for more information.

Starting and Ending a Linux Session

Starting a Linux session. To start a Linux session, type in your username and password at the welcome screen. Do not share your password with anyone, else others could potentially make mischief, either intentionally or unintentionally, in your work area. The system administrator will set up an account for you, and help you configure your desktop, which will look something like Figure 6.

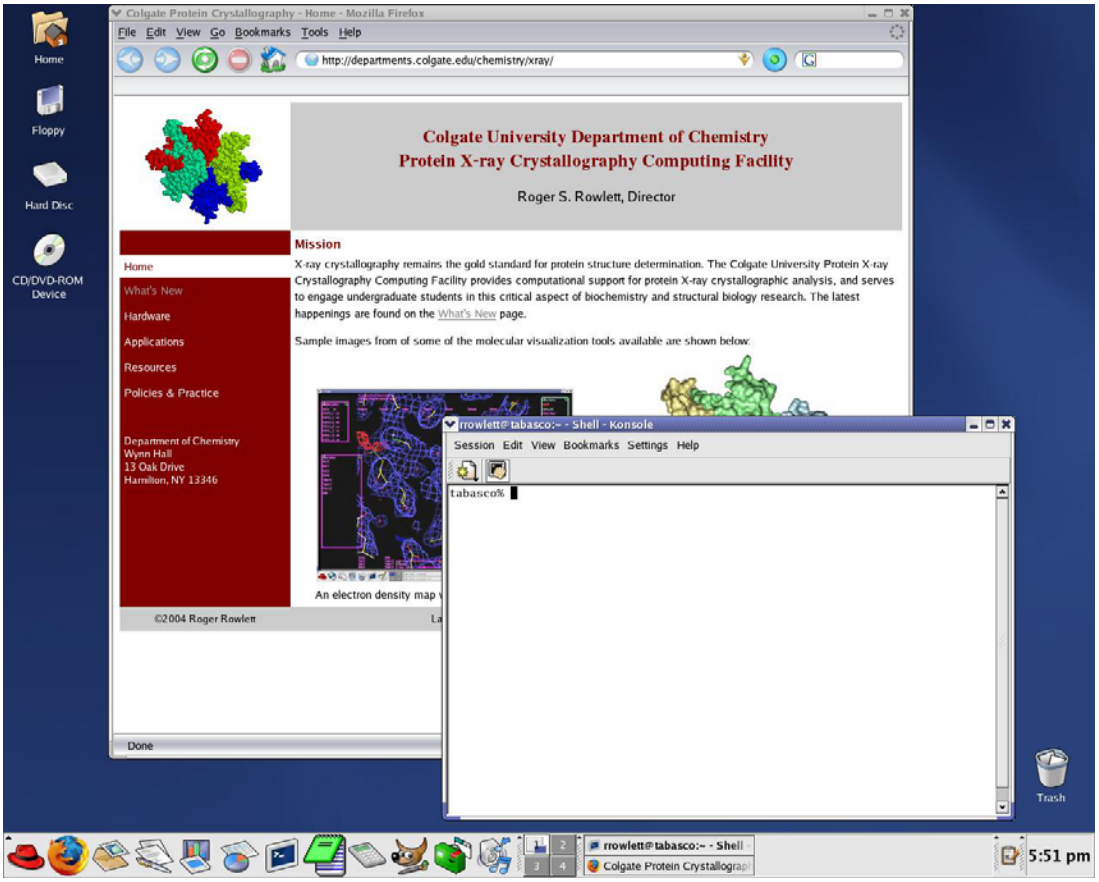

Home
Floppy
Hard Disc
CD/DVD-ROM Device
Colgate Protein Crystallography - Home - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://departments.colgate.edu/chemistry/xray/
Colgate University Department of Chemistry
Protein X-ray Crystallography Computing Facility
Roger S. Rowlett, Director
Mission
X-ray crystallography remains the gold standard for protein structure determination. The Colgate University Protein X-ray Crystallography Computing Facility provides computational support for protein X-ray crystallographic analysis, and serves to engage undergraduate students in this critical aspect of biochemistry and structural biology research. The latest happenings are found on the What's New page.
Sample images from of some of the molecular visualization tools available are shown below:
Home
What's New
Hardware
Applications
Resources
Policies & Practice
Department of Chemistry
Wynn Hall
13 Oak Drive
Hamilton, NY 13346
©2004 Roger Rowlett
An electron density map
rrowlett@tabasco:~ - Shell - Konsole
Session Edit View Bookmarks Settings Help
tabasco%
Done
Trash
rrowlett@tabasco:~ - Shell
Colgate Protein Crystallograp
5:51 pm

Figure 6. Typical Linux session (using KDE desktop) on a workstation in the Colgate Protein X-ray Crystallography Computing Facility. The Firefox browser and a terminal window are open on the desktop in this figure.

Starting a Linux Shell. Most crystallography programs and utilities run from a shell window, which is basically a text window into which you can type Linux commands. To open a new shell in Linux click on the terminal icon in the toobar (7th from the left in Figure 6). A new window should open with a command prompt such as `ancho%`. In this case, `ancho` indicates the computer to which you are logged in, and `%` is the shell prompt. Commands that you type will appear after the prompt symbol. Shell windows can be closed by typing `exit` at the prompt or by clicking the upper right corner of the window.

Ending a Linux session. To end a Linux session (not to be confused with `exit` from a shell window), right-click on the desktop and select *Logout*. Always log out of your session when you are away from your terminal for more than a few minutes.

File Storage Structure in Unix/Linux

File directory structures are similar to that of DOS (the precursor to Windows). Indeed, DOS (now called the Command Prompt in Windows) is a derivative of Unix and shares many common commands and functions. When you start a Linux session, you will be located in your home directory, and all commands you type will normally apply to the files in this, your local directory, unless instructed otherwise. Your local directory might be something like `/home/jdoe`. That is, unless instructed otherwise, all files will be read and written to the `jdoe` directory of the `home` directory of the machine you are logged in. You can find out where you currently are by typing the `pwd` command; you can make a new directory in the current directory with the `mkdir` command; or you can change your local directory to another with the `cd` command. These and other commands are described below.

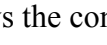
The string `/home/jdoe/filename` describes an *absolute path* to the file `filename`, a complete set of instructions to locate the file in question. The leading slash indicates that this is a complete path, starting with the directory `home`. The string `datafiles/filename` is a *relative path* which describes how to locate a file from the local directory. A relative path does not start with a leading slash. For example if you were currently in the directory `/home/jdoe`, the relative path `datafiles/filename` would point to the absolute location `/home/jdoe/datafiles/filename`. Relative paths can save a lot of time when typing commands.

It is important for new Linux users to know that Linux will not generally protect you from yourself. For example, deleting files or directories in Linux is absolutely, positively, no-turning-back, irretrievably *FINAL*. You cannot recover files you delete accidentally. Therefore, proceed with care and caution when cleaning up data. A list of commonly used Linux commands follows in the next section.

Commonly Used Linux Commands

The following is an alphabetical list of a common Unix commands that you might use for routine crystallography work and file maintenance. Please note that Linux commands, unlike DOS commands, are case-sensitive. So `PWD` is not the same as `Pwd` as `pwd`. Filenames are also case-sensitive; most users avoid using capitalized text in filenames for ease of typing and to prevent confusion.

- `cd` *directory*—change your local directory to a new location. If you issue `cd` with no argument, it will take you to your home directory.

- `chmod` *permission filename*—change permissions for files. You must provide a filename and one or more permission arguments with this command. The permission argument includes an optional group (`u`ser, `g`roup, `o`thers, `a`ll) and a permission (`e`xecute, `r`ead, `w`rite) For example to allow a file to be executable, you would type `chmod +x filename`. To make a file readable and executable by all users, but *not* writable, you would type `chmod a+rx-w filename`. This command is most often used to make scripts you write executable, which is not the default. For example `chmod +x filename` would make a file executable for the user, in addition to whatever permissions it already had.
- `cp` *filename1 filename2*—copies a file from one location to another. For example `cp /home/jdoe/yourfile myfile` would copy the file `yourfile` from the `/home/jdoe` directory into your local directory with the name `myfile`. Be careful with the `cp` command: it will not check to see if you are copying over a current file with the same name.
- `df`—report free block of space on disk drive. To force display of free space in intelligible units (kB and MB), use `df -h.`
- `kill` *PID*—halts a process with the indicated PID (process identification number). This command is used to halt a program running in the background. Sometimes kill is not adequate and a more severe variant `kill -9 PID` must be used. The `kill -9` command should normally be used only as a last resort.
- `ls`—list directory. This command will list the contents of the current directory. You may add switches for additional functionality. For example `ls -l` will make a "long" listing of files with additional file details, including size and permissions. On most Linux distributions, `ll` will carry out the `ls -l` command.
- `mkdir` *directory*—creates a new subdirectory within the local directory.
- `less` *filename*—displays the contents of a file a page at a time. Tapping  or scrolls one page forward, one page backward; jumps to the beginning of the file; jumps to the end of the file. Type to quit.
- `mv` *filename1 filename2*— moves a file from one location to another. For example `mv /home/jdoe/yourfile myfile` would move the file `yourfile` from the `/home/jdoe` directory into your local directory with the name `myfile`. Be careful with the `mv` command: it will not check to see if you are copying over a current file with the same name. The mv command is often used to rename a file in the local directory. For example `mv oldname newname` would rename the file `oldname` to `newname`.
- `kedit` *filename*—KEdit is a very nice KDE text editor that can be used to edit files and scripts. If a filename is supplied it will open that file. An alternative editor, if installed, is `editpad`.
- `ps`—identifies the current processes running and their process identification numbers (PID). This command is most often used to obtain PIDs for the `kill` command.
- `pwd`—print working directory. This command returns the name of the directory you are currently located in.
- `rm` *filename*—deletes a file from the local directory. A useful but very dangerous variant of remove is `rm -r` *directory*. The `rm -r` command is a recursive remove which

deletes a directory and absolutely everything that is in it, including additional subdirectories and files within it. Use with extreme caution!

- `rmdir` *directory*—removes a subdirectory from within the local directory. The subdirectory must be empty of files in order to remove it.
- `tail` *n filename*—displays the last *n* lines of the file *filename*. A variant, `tail –f` *filename* will continuously follow the last lines of a file as it is being written. This command is useful for monitoring the progress of programs that write log files while executing. The `tail –f` command must be terminated with .

Special Linux Command Line Characters and Actions

Linux has many special characters that make it easier to type commands. Some of these are listed below, with examples.

- The tilde (`~`) is used to designate your home directory. For example, if your home directory is `home/jdoe`, then the command `ls ~/datafiles` would list the contents of the `home/jdoe/datafiles` directory.
- The dot (`.`) is used to designate the current local directory. For example, the command `cp /home/jdoe/datafiles/filename .` would copy the file `filename` from the `/home/jdoe/datafiles` directory into your current directory using the same name.
- The double dot (`..`) is used to designate the directory one level up from your current directory. For example the command `cp ../../datafiles/filename .` would copy the file `filename` from the directory two levels up into your current directory.
- The star (`*`) is a wildcard character that can be used to select many similar files. For example the command `cp /home/jdoe/datafiles/*.osc .` would copy all files in the `/home/jdoe/datafiles` directory ending with the characters `.osc` into your current directory. Be careful with wildcards, especially when removing files. You can always test your wildcard selection by doing an `ls` command first. If the `ls` command lists the files you thought you selected, you can change `ls` to `rm` and remove the correct files with confidence.
- The question mark (`?`) is a wildcard character for a single character in a filename. For example, the command `rm abcd?` would remove from the current directory all files exactly five letters long starting with the letters `abcd` and any other fifth character.
- Brackets (`[]`) are used to enclose ranges of characters allowable in a single character position for selected files. For example the command `rm mydata.1[0-5][0-9].osc` would remove files `mydata.100` to `mydata.159` if present in the current directory.
- The ampersand (`&`) is used to instruct Linux to run a program in the background. In this way, you can continue to use the current Linux shell while your program runs. Background jobs will continue to run even if you log off the machine. For example the command `myprog &` would start the program `myprog`, display a PID, and return the shell prompt. The program will run in the background until it finishes or is terminated with the `kill` command.

- The up-arrow key (↑) will display the last command typed if your environment is set up appropriately. Repeatedly pressing ↑ will call up additional previous commands. Pressing the ↓ key will bring up successively more recent commands. Commands that are called up this way can be edited, using the ← and → keys to scroll across the line. To execute a command called up and/or edited this way, press ENTER.
- The middle mouse button actually has an important use in Linux as a "paste" command. This is especially useful when editing command lines with long file names. You can select text in virtually any Linux application, including the terminal window, by holding down the left mouse button and dragging. To paste this text into the command line (or another Linux application) move the cursor to the insertion point and click the middle mouse button.

Input and Ouput Redirection

Linux allows the user to redirect information from the keyboard or screen (defaults for input and output) to files or even other programs using redirection commands. A listing of common redirection commands is given below with examples.

- The left carat (<) is used to redirect input. For example, the command `myprog<input.txt` would launch the program `myprog` and accept input from the text file `input.txt` rather than the (default) keyboard. Running programs using input scripts rather than the keyboard is a very common way of executing programs in Linux.
- The right carat (>) is used to redirect output. For example, the command `myprog<input.txt>output.txt &` would launch the program `myprog` in the background, accept input from the text file `input.txt` rather than the (default) keyboard, and output results to the file `output.txt` rather than the (default) screen. You could monitor the progress of `myprog` if desired by issuing the command `tail –f output.txt`.
- The double right carat (>>) is like the right carat except that it will append, rather that overwrite, data to an output file. For example the command `myprog<input.txt>>output.txt &` would launch the program `myprog` in the background, accept input from the text file `input.txt` rather than the (default) keyboard, and append results to the file `output.txt` rather than the (default) screen. If `output.txt` does not already exist, it will be created.
- The pipe (|) is used to feed the output of one program into another. For example, the command `myprog<input.txt|tee output.txt` would launch the program `myprog`, accept input from the file `input.txt`, and send the results to the program tee, which sends output to both the screen and the file `output.txt`. This example is another way to monitor the progress of an executing program while saving the output to a text file.

Customizing Your Linux Environment

It is possible to customize your Linux environment to make it easier to navigate through your directories and projects. To customize your environment, edit the `.tcshrc` file in your home directory. Commands in this directory will be executed each time you open a new shell window. The following types of commands are useful to have in your `.tcshrc` file:

- `set history = 100`—this setting allows the last 100 commands to be remembered. You can call them up at the prompt by pressing the key as described previously.
- `alias` *name* `'command'`—This command is used to designate a shortcut name for a complicated command. For example the command `alias project10 'cd /projects/project10/refine/ncs/'` would allow you execute the long complicated directory change in single quotes by simply typing `project10` at the prompt. Any valid Linux command can be placed within the single quotes.

Additional Resources

- Beginner's Linux Guide: http://www.linux.ie/newusers/beginners-linux-guide/
- A Beginner's Guide to Linux: http://www.geocities.com/aboutlinux/

X-ray Diffraction Data Analysis

There are many software suites that can be used to analyze protein X-ray diffraction data. Described here is data analysis procedures based on the programs DENZO and SCALEPACK or MOSFLM and SCALA.

Indexing the first frame using DENZO

When screening crystals for diffraction, or before analyzing an entire data set, it is necessary to index the reflections observed in the first frame in order to obtain the exact orientation of the crystal with respect to the X-ray beam, and to make a preliminary determination of the space group and unit cell size. The programs XDISP and DENZO are used together to perform reflection indexing. The following procedure is typical for indexing a single frame of data.

- Copy your image file(s) (typically *.osc) into a local working directory
- Display a frame by issuing the command `xdisp raxis4 100 filename.osc &`, where *filename.osc* is the name of your image file.[5] An image of the diffraction pattern should appear. If it looks usable, proceed on to the next step.
- Obtain and edit the file `index` (file 1), which contains instructions for DENZO indexing. This file should reside in the same directory as your image files. You should pay special attention to the following settings in the file, and change them as necessary:
    - `wavelength`—if not using a rotating copper anode source (1.5418 Å) change to the appropriate value
    - `x-beam and y-beam`—obtain the correct values for the X-ray beam center from the latest log book entry for the instrument. It is difficult to properly index reflections unless these values are accurately known.
    - `distance`—enter the camera distance here. This value must be known accurately in order to properly index reflections.
    - `mosaicity`—a measure of the disorder of the crystalline lattice. Start with a value of 0.4–0.7 degrees
    - `raw data file`—enter the image file name here, with the # symbol used to indicate the number and position of the numerical part of the filename that tracks the frame number.
    - `space group`—if known, enter the space group here; if unknown, start with the lowest symmetry space group, P1.
    - `oscillation range`—enter the oscillation angle used here, typically 0.2–2.0°. This value must be known accurately in order to properly index reflections.
    - `sector`—enter the frame number of the file you would like to index. Normally it should be the same file that you are displaying in XDISP.
    - `box print`—defines the background area around each spot. Set this value to about 3 times the radius of the spot size.

[5] The command as written here is appropriate for images collected by an RAXIS-IV system with 100 μ*M* resolution image plates. The format of this command may be slightly different depending on the detector system used.

  - `spot radius`—enter the spot size in mm here. The spot radius should be large enough completely enclose desired spots, but not so large that spots overlap. Start with a value of 0.7 mm.
  - `background radius`—defines a buffer zone between the spot and the background. Typically set to 0.1 mm larger than the spot size. If this line is omitted, DENZO will assign the bare minimum buffer around the spot.
- Before attempting to index the frame using DENZO, identify major peaks in the image by pressing the *Peak Sear* button in XDISP.
- Index the frame by entering the command `denzo<index>index.out` and observe the XDISP window. If things are working properly, the peaks identified by the peak search should turn green, and as refinement proceeds, yellow.
- Open a zoom window by pressing the *Zoom Wind* button in XDISP. The position of the zoom window in the main frame can be changed by pointing with the mouse and clicking the middle button.
- In the zoom window, press the *Int. Box* button so that you can observe the spot size and integration boxes. Examine all around the diffraction pattern using the zoom window. In particular,
  - Verify that the predicted spots match the observed spots. If the preds are badly mismatched, the most likely culprits are incorrect x-beam/y-beam values, incorrect distance, or incorrect space group. If the space group is suspect, re-index using space group P1.
  - If the spots index well, but you have more preds than spots, then the mosacity is too high. Lower the mosacity and re-index. If you have more spots than preds, then the mosaicity is probably too low. Increase mosacity and re-index.
  - Verify that the spot size is sufficient to enclose desired spots. If spot size is too large, many reflections will be rejected due to overlap with other spots. If spot size is too small, reflections will be rejected for spilling over into the background.
- Examine the log file `index.out` to determine the quality of the fit and verify space group assignment. In particular,
  - Examine the χ^2 value for the x-direction, y-direction and partials. χ^2 values near 1 indicate an acceptable fit. χ^2 values over 4 should be cause for concern.
  - Examine the space group fitting statistics. The correct Bravais lattice space group is most likely to be the highest symmetry space group with a distortion index <0.5%
- Do not delete the `index.out` file. You will need the last 20 lines of this file to properly orient DENZO for the fitting of your entire data set, as described below.

File 1
index

```
title 'DENZO autoindexing'

[Detector Information]
format raxis4 100
use beam
wavelength 1.5418
error density 0.6
error systematic 5.0 partiality 0.15 positional 0.050
weak level 5.0
film rotation 180.0
x beam  150.2 y beam  151.6
y scale 1.0
skew 0.000
cassette rotx 0. roty 0.
profile fitting radius 25.0
resolution limits 30.0 2.0
distance 180.0

[Fitting Information]
mosaicity 0.7
raw data file   'hica15-####.osc'
space group C2
oscillation range 0.5  start 0.0
sector 1
box print 2.5 2.5
spot radius 0.8
background radius 0.9
overlap spot

[Refinement]
fix x beam y beam
fit cell
fit crystal rotx roty rotz
go go go go
fit all
fix y scale skew
fix x beam y beam
go go go go go go go
go go go go go go go
go go go go go go go
go go go go go go go
write predictions
print statistics
peak search file 'peaks.file'
go go go go go go go
go go go go go go go
go go go go go go go
go go go go go go go
write predictions
go go go go go go go go go go
go go go go go go go
go go go go go go go
fit all
go go go go go go go
go go go go go go go
go go go go go go go
list
```

Integrating the entire data set using DENZO

Once you have determined the correct parameters for indexing your first frame, including the space group, you can integrate a series of frames or a complete data set to catalog all of the observed reflections. The programs XDISP and DENZO are used together to perform reflection integration. The following procedure is typical for integrating data.

- Copy your image file(s) (typically *.osc) into a local working directory.
- Create a subdirectory `x` to receive the integrated reflection intensities.
- Display the first frame by issuing the command `xdisp raxis4 100 filename.osc &`, where `filename.osc` is the name of the first image file.[6] An image of the diffraction pattern should appear.
- Obtain and edit the file `integrate` (file 2), which contains instructions for DENZO integration. This file should reside in the same directory as your images. You should:
    - Copy the last 20 lines of the previously saved `index.out` file into the appropriate part of the integrate file. These lines contain information about the crystal orientation and refined instrumental parameters in the first frame. Delete any lines from this section that duplicate other instructions in the integrate file, *e.g.*, mosaicity settings.
    - Enter the directory and filename for the reflection files in the `film output` line.
    - Choose the frames you would like to integrate by editing the `sector` instruction.
    - Other settings should be edited to match those used to index the first frame.
- Before attempting to integrate using DENZO, identify major peaks in the first frame by pressing the *Peak Sear* button in XDISP.
- Integrate the data by entering the command `denzo<integrate>integrate.out &` and observe the XDISP window. If things are working properly, the peaks identified by the peak search should turn yellow as they are identified; overloads, peak overlaps, and spots too large to fit in the current spot size are flagged in red.
- Observe the output stream from DENZO in real time by typing the command `tail –f integrate.out`. This will allow you to examine the output as integration proceeds.
- Open a zoom window by pressing the *Zoom Wind* button in XDISP. The position of the zoom window in the main frame can be changed by pointing with the mouse and clicking the middle button. Place this window on a convenient portion of the image so that you can follow the progress of the integration.

[6] The command as written here is appropriate for images collected by an RAXIS-IV system with 100 μ*M* resolution image plates. The format of this command may be slightly different depending on the detector system used.

File 2
integrate

```
title  'Refine All Images'
format raxis4 100

[*****INSERT CRYSTAL ORIENTATION PARAMTERS HERE*****]
cassette rotx   -0.01 roty   -0.07 rotz    0.00 2 theta    0.00
distance  179.33
x beam  150.473 y beam  151.274
y scale  1.00340
film rotation  180.000
skew  0.00025
crossfire y  0.001 x  0.051 xy -0.043
goniostat single axis
goniostat orientation    0.000    0.000
motor axis  0.000000  1.000000  0.000000
profile fitting radius   25.00
resolution limits   30.0   2.00
wavelength 1.54180
monochromator   0.000
spindle axis   0   0   1 vertical axis   1   0   0
oscillation start    0.00 end    0.50
unit cell  232.649  144.639   52.101   90.000   94.104   90.000
crystal rotx -111.721 roty   23.818 rotz -132.990
[*****END of CRYSTAL ORIENTATION PARAMETERS*****]

[Fitting Parameters]
resolution limits 25.0 2.0
space group C2


mosaicity 0.60
oscillation range 0.5 start 0.0
sector 1 to 280
error density 0.6
error systematic 5.0 partiality 0.10 positional 0.070
weak level 5.0
profile fitting radius 25.0
raw data file 'hica15-####.osc'
film output file 'x/hica15-####.x' [***create directory 'x' prior to run***]
box 2.5 2.5
spot radius 0.8
background radius 0.9
overlap spot

[Refinement Begins]
start refinement
print no profiles
fit crystal rotx roty rotz
go go go go go go
go go go go go go
write predictions

fit cell
go go go
go go go go go go
go go go go go go

fit all fix y scale skew
go go go go go go
go go go go go go
go go go go go go
list
print profiles 1 1
calculate go
end of pack
end of job
```

- In the zoom window, press the *Int. Box* button so that you can observe the spot size and integration boxes. Examine the diffraction pattern as it is integrated. If everything is working properly, the preds should match with the observed spots from frame to frame, and there should normally be about as many preds as spots.[7] If overlaps (red spots) exceed 10% of the total, the data is likely to be unusable; you will have to choose different integration conditions or change your data collection parameters to generate better-separated reflections.
- Examine the `integrate.out` file as it streams out. In particular,
    - Examine the mosaicity histogram for each frame. If the mosaicity is properly set, most of the bins should be filled, and should decrease smoothly as you read down the screen. If the histogram is jagged or non-monotonic, there is a problem with the mosaicity setting, or worse.
    - Examine the χ^2 for the fit on each frame. Ideally, it should be near a value of 1 for x-, y-, and partials. Values >4 are cause for concern.
- If all has gone well, you can proceed to data scaling, as described in the next section

Scaling Reflection Data using SCALEPACK

Before integrated reflection data can be used to produce an electron density map, it must be scaled and evaluated for completeness and quality. Scaling is done by a stand-alone program SCALEPACK. The following procedure is typical for scaling data:

- Obtain and edit the file `scale` (file 3), which contains SCALEPACK instructions for scaling your reflection data. You should:
    - Set the `estimated error` for each of the bins to ≈0.05
    - Set the `error scale` multiplier to ≈ 2.0
    - Choose the frames you would like to scale in the `sector` line and all of the `fit` and `add partial` lines.
    - Choose the resolution range of the scaled data in the `resolution` line.
    - Edit the filename for the appropriate reflection files on the `file` line.
    - Choose an `output file` name, typically *filename.sca*.
- Obtain a copy of the file `scalepack-stats` and place it in the same directory as `scale`. This utility organizes the output of SCALEPACK into a handy format.
- Obtain a copy of `runscale` and place it in the same directory as `scale`. `Runscale` is a script file to run SCALEPACK using `scale` as input and filters output through `scalepack-stats` to display a quick summary of the scaling operation.
- To begin scaling, issue the command `runscale`. When scaling is completed a summary of the scaling operation will be displayed. The summarized statistics can also be viewed in the file `scale-stats`.

[7] It is generally better to err on the side of too many preds (higher mosaicity) than too few (lower mosaicity), else reflections may be missed.

File 3
`scale`

```
scalepack << eof-scalepack
format denzo_ip
number of zones 10
estimated error  0.044 0.052 0.065 0.075 0.085 0.11 0.13 0.15 0.18 0.23
error scale factor 1.8
rejection probability 1.e-4
write rejection file 0.5
sigma cutoff -3.0
postrefine  6
@reject
scale restrain 0.05
b restrain 0.2
ignore overloads
       [Crystal data]
space group C2
       [Define images]
reference film 1

resolution 30 2.2
sector 1 to 280
FILE 1 'x/hica15-####.x'

       [Control postrefinement]

 fit crystal a* 1 to 280
 fit crystal b* 1 to 280
 fit crystal c* 1 to 280
 fit crystal alpha* 1 to 280
 fit crystal beta*  1 to 280
 fit crystal gamma* 1 to 280
 fit film rotx    1 to 280
 fit film roty    1 to 280
 fit crystal mosxx   1 to 280

       [Output]
add partials   1 to 280
output file  hica15.sca
[output anomalous]

eof-scalepack
```

- Examine the scaling statistics, paying particular attention to the following items:
  - The overall R_{sym} should be very low, typically ≈0.05 for an excellent data set. Overall R_{sym} values >0.10 are cause for concern. For a typical data set the R_{sym} values by shell should monotonically increase from low- to high-resolution shells. You should probably disregard high-resolution data with R_{sym}>0.30, and should re-adjust `resolution` appropriately in the `scale` file.
  - The overall I/σ(I) value should typically be ≈20. An overall I/σ(I) value <10 is cause for concern. For a typical data set the I/σ(I) value should monotonically decrease from low- to high-resolution shells. You should probably disregard shells with I/σ(I) values <2, and should re-adjust `resolution` appropriately in the `scale` file.
  - Examine the overall % completeness. Data that is 85-90% complete should be sufficient to solve a structure, although more completeness is better if practical. A quality data set will also have approximately the same level of completeness in each shell, perhaps with a monotonic fall-off at high resolution where spot

intensities get weaker. Gaps in completeness in low- or middle-resolution shells may indicate problems with ice rings and/or integration.

- If you are scaling the final data set in preparation for producing an electron density map, you should examine the χ^2 values. In a properly scaled data set the overall χ^2 and the χ^2 for the individual data shells should be 1.00 ± 0.02. The χ^2 can be adjusted for each shell as follows:
    - If all χ^2 values are too high, adjust by raising the value of `error scale factor.` If all χ^2 values are too low, adjust by lowering the value of `error scale factor.`
    - To adjust χ^2 in individual resolution bins, change the value of `estimated error` for that bin. If χ^2 is too high, raise the value of `estimated error`; If χ^2 is too low, lower the value of `estimated error`.
    - Rescale data, adjusting error scale factory and/or estimated error until all resolution bins have a χ^2 value of 1.00 ± 0.02 and the overall χ^2 is close to 1.0
- The output reflection (*.sca) file is the raw data from which the data will be solved. If this file is satisfactory, the image files may be compressed, backed up, and removed from computer to save space.

Indexing the first frame in MOSFLM

When screening crystals for diffraction, or before analyzing an entire data set, it is necessary to index the reflections observed in the first frame in order to obtain the exact orientation of the crystal with respect to the X-ray beam, and to make a preliminary determination of the space group and unit cell dimensions. The program MOSFLM, which is part of the CCP4 suite of protein crystallography programs, can be used to perform reflection indexing. The following procedure is typical for indexing a single frame of data.

- Copy your image files(s) (typically *.img) into a local working directory.
- Navigate to this directory and configure CCP4 by issuing the command `ccp4setup`, then start MOSFLM by issuing the command `ipmosflm`. The MOSFLM prompt should appear.
- Load an image into MOSFLM using the `image` and `go` commands. For example, to load the image file `xyz-001.img` issue the commands `image xyz-001.img` then `go`. The MOSFLM graphical window should appear with your loaded image file (Figure 7). For most image file formats, MOSFLM can read camera distance and X-ray wavelength data directly from the file. Check the Processing Parameters pane to verify these values are correct. Processing parameters can be changed by clicking on the item you want to change and typing the desired value.
- Enter accurate values for Beam X and Beam Y in the Processing Pane by clicking on these items and typing the desired value
- Before indexing, it is helpful to set the detector gain and backstop radius. This is conveniently done by selecting *Keyword input* on the main menu. Keyword input is terminated by typing `end` after the last keyword entry. The backstop radius should be just large enough to exclude the central region of the image blocked by the beam stop; the detector gain should be set to a value suggested by the manufacturer. The following

keyword commands are typical for images derived from the Oxford Diffraction Excalibur system:

```
o backstop radius 4.0
o gain 1.2
o end
```

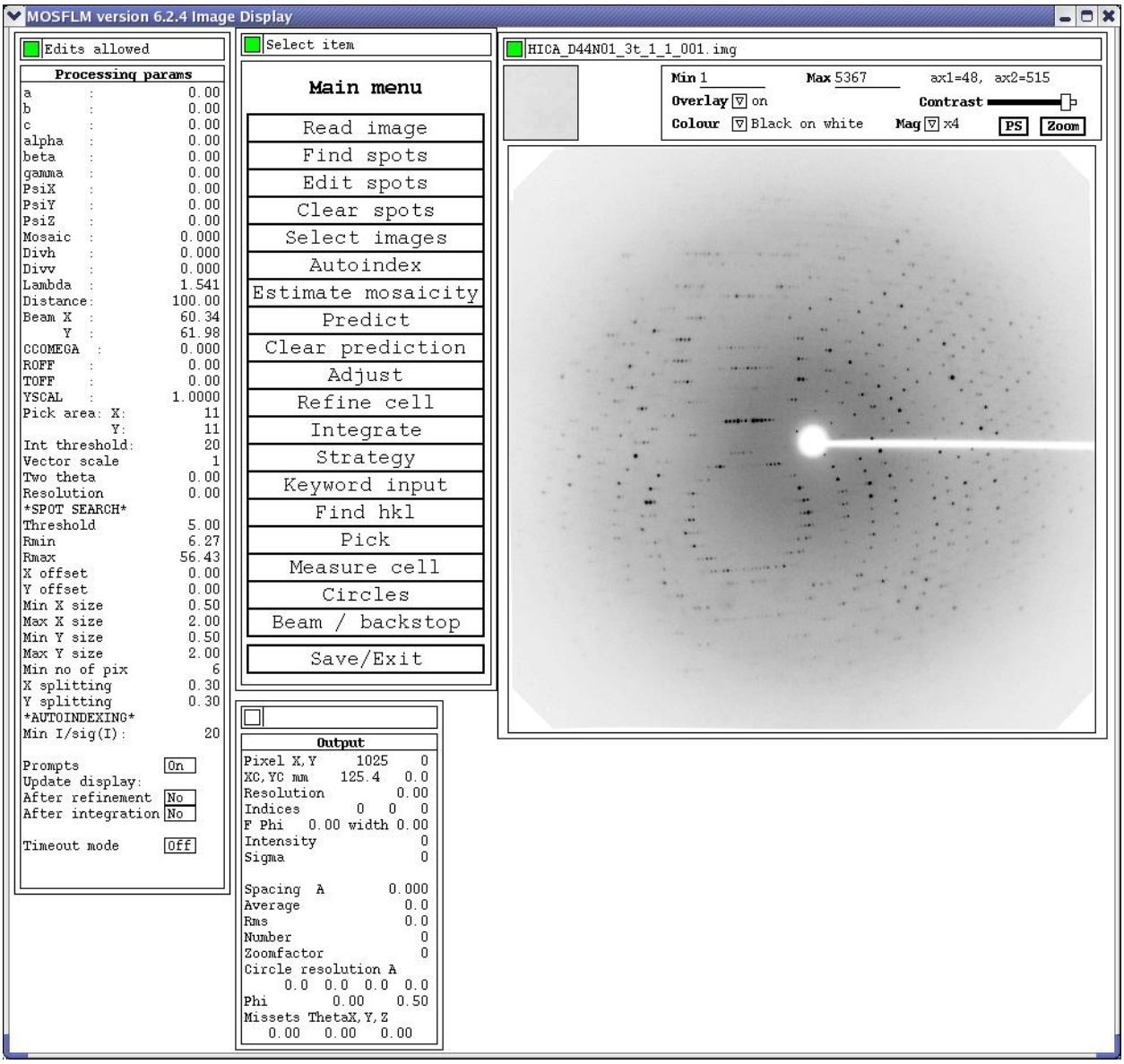


Figure 7. MOSFLM graphical window. Processing parameters are displayed on the left; main menu is in the middle; image pane is on the right.

- Commence autoindexing by clicking on *Autoindex* in the main menu. You will be prompted with a series of questions, shown below with typical answers. Comments are given in italics.

`Do you wish to continue? Y` (*the default can always be accepted by typing*)

`Do you want to find spots manually? N`
`Do you want to add spots manually? N`
`Do you want to try the new autoindexing? Y`
`Do you want to fix detector distance? Y`
`Do you want to exclude spots close to ice-rings? N`
(*applicable only if you have ice rings*)
`Filename for final orientation matrix: xyz.mat`
(*enter a filename of your choice*)
`Maximum expected cell edge (Angstroms): 190`
(*choose a value that you believe is larger than your longest dimension; if autoindexing fails, you might try autoindexing again with a larger value*)
`Do you want to pre-refine the solutions? N`
`Do you want to proceed? Y`
`Select a solution AND a spacegroup from list: 10 P41212`
(*A list of possible space groups will be listed, along with their penalty functions. Normally you should choose the highest symmetry space group that has a low penalty score. You should notice a large gap in penalty scores between acceptable and unacceptable space groups. MOSFLM will suggest the best solution for you.*)
`Positional sigma cutoff [2.50]:` (*accept the default*)
`Do you want to update cell parameters: Y`
`Do you want to accept the new beam coordinates? Y`
`Do you want to accept this solution? Y`

- You should observe a series of red crosses in the graphical window indicating spots that were used to perform the autoindexing. Clear these from the image by clicking on *Clear spots* in the main menu.
- Call up spot predictions by clicking on *Predict* in the main menu. You will observe a series of colored boxes in the graphical window. These boxes should correspond to the positions of spots in the image.
- If not all the spots on the image are accurately predicted, the mosaicity should be adjusted in the processing parameters pane by clicking on `mosaic` and changing the value. Typical flash-frozen crystals have mosaicity values between 0.2° and 0.8°. Experiment with various values for the mosaicity until the spots are accurately predicted. If you have more spots than predictions, increase the mosaicity; if you have more predictions than spots, decrease the mosaicity. You need not be too fine here: the mosaicity value will be refined by MOSFLM during integration.
- Good agreement between spots and predictions is a prerequisite for integration of the full data set.

Integrating the entire data set using MOSFLM

Once you have determined the correct parameters for indexing your first image, you can refine the unit cell parameters and then integrate a series of frames or a complete data set to catalog all the observed reflections and their intensities. In MOSFLM, this is conveniently done immediately after autoindexing the first frame of data, without closing your MOSFLM session.

If you are integrating a data set from scratch, you may want to re-run autoindexing before proceeding. The following procedure is typical for data set integration:

- It is important to set a few keywords before proceeding to make the data easier to process in SCALA later. Click on *Keyword input* and set the following keywords:
  - `PNAME XYZ01` (*enter a project name*)
  - `XNAME XYZ01` (*enter a crystal name*)
  - `DNAME HighRes` (*enter a dataset name*)
- Additional keywords should be entered at this time, including information about the resolution range to be processed, and the expected separation between spots in mm:
  - `Resolution 30 2.8` (*enter resolution range of reflections to be processed*)
  - `Separation 0.9 0.9` (*this keyword is especially useful for data from large unit cells where reflection spacing is tight. Lower values allow recognition and quantification of spots closer together. Use with care: verify that integration boxes are large enough to accommodate entire spots when reducing separation.*)
  - `end`
- Refine unit cell parameters by clicking on *Refine cell* in the main menu. You will be prompted with a series of questions, shown below with typical answers. Comments are given in italics.

```
Give number of segments: 2
```
(*it is generally best to refine cell parameters with two segments of images*)
```
Image number for first image of segment 1: 1
```
(*enter image number*)
```
Image identifier: XYZ
```
(enter filename prefix without image number or accept default)
```
Use phi values from image header? Y
Number of images in this segment? 4
```
(*choose 3-4 images*)
```
Use the current crystal orientation? Y
Image number of first segment of segment 2? 90
```
(*you should choose a second segment that is separated from the first segment by 45-90° in phi*)
```
Filename for final orientation matrix: XYZ-highres.mat
```
(*enter filename of your choice*)
```
Do you want to proceed? Y
```
(*cell refinement proceeds ;this will take a few minutes*)
```
Reset missets to those of the first image? Y
```

- When cell refinement is completed, turn on predictions by clicking on *Predict* in the main menu. Spots should be accurately predicted.
- Integration can be commenced by clicking on *Integrate* in the main menu. If desired, additional keywords may be entered via the *Keyword input* item on the main menu before proceeding. You will be prompted with a series of questions, shown below with typical answers. Comments are given in italics.

```
Do you want to update any of these? N
```
(*not necessary unless analyzing synchrotron data*)

```
Give first, last image numbers: 1 166 (enter image data range)
Use phi values from image header? Y
Give BLOCK and/or ADD keywords if required: (press ENTER)
Refine cell parameters? N
Write a new MTZ file for each block of data? N
MTZ filename? XYZ-highres.mtz
      (enter filename of your choice with .mtz extension)
Do you want to proceed? Y
```

- Exit the MOSFLM graphics window by clicking on *Save/Exit* in the main menu. *Do not close MOSFLM by clicking on the upper right hand corner of the MOSFLM graphics window! The program will crash!*
- Exit MOSFLM by typing `exit` at the MOSFLM prompt.

Scaling reflection data using SCALA

If integration has gone well, you can proceed to scaling data using SCALA. The most convenient way to use SCALA is through the CCP4i interface. In general the SCALA default settings are very good, and scaling of data is quite transparent. The following procedure is typical:

- Configure CCP4 by typing `ccp4setup` at the prompt. Start CCP4i by issuing the command `ccp4i` at the prompt. The CCP4i graphical interface will open (Figure 8).

CCP4 Program Suite 5.0.2 CCP4Interface 1.3.20 running on tabasco.colgate.edu Project: d44n01
List of jobs (finished or running) in this project
Help
Data Reduction
Integrate Images
Import Merged Data
Import Unmerged Data
Sort/Modify/Combine MTZ Files
Scale and Merge Intensities
Convert Intensities to SFs
Treat Twinned Data
6 15 Feb 05 FINISHED export mtz2fobs
5 15 Feb 05 FAILED export mtz2fobs
4 15 Feb 05 FINISHED scala scalemerge
3 15 Feb 05 FINISHED scala scalemerge
2 15 Feb 05 FINISHED sortmtz combine
1 15 Feb 05 FINISHED sortmtz renumber
List of jobs (finished or running) in this project
Directories&ProjectDir
View Any File
View Files from Job
Delete/Archive Files..
Kill Job
ReRun Job..
Edit Job Data
Preferences
System Administration
Mail CCP4
Exit

Figure 8. Main task window for CCP4i. Tasks are listed in the left pane, jobs in the middle pane, and administration functions in the right pane.

- If you have not already done so, set up and select a project directory by clicking on *Directories&ProjectDir* in the administration pane.
- Select the *Data Reduction* module in CCP4i (upper left menu bar) and click on *Scale and Merge Intensities*. A task window will open (Figure 9.) You will need to enter a job title, select the appropriate MTZ file to be scaled (from your MOSFLM integration), define an output MTZ filename (different from the input MTZ filename) and (optionally) the

estimated number of residues in the asymmetric unit. The latter is useful if you would like to obtain an estimated average *b*-factor for the data set from a Wilson plot. If PNAME, XNAME, and DNAME were set in MOSFLM before integration, these will be successfully read into the job under *Define Output Datasets*. These parameters are mandatory if you are merging two or more datasets together in CCP4i. The *Scaling Protocol* defaults should be fine in 99% of cases.

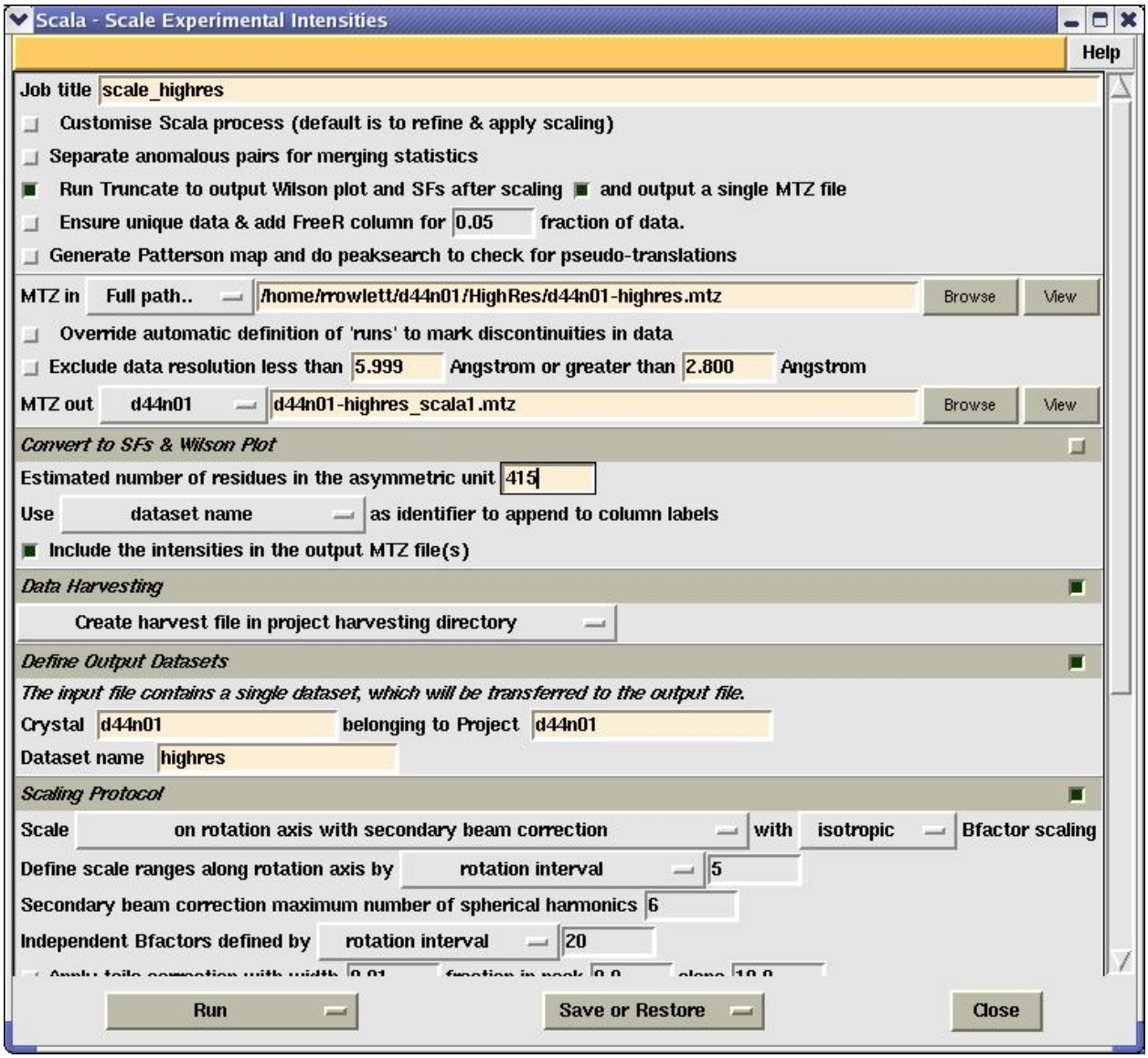

Scala - Scale Experimental Intensities
Help
Job title scale_highres
Customise Scala process (default is to refine & apply scaling)
Separate anomalous pairs for merging statistics
Run Truncate to output Wilson plot and SFs after scaling and output a single MTZ file
Ensure unique data & add FreeR column for 0.05 fraction of data.
Generate Patterson map and do peaksearch to check for pseudo-translations
MTZ in Full path.. /home/rrowlett/d44n01/HighRes/d44n01-highres.mtz Browse View
Override automatic definition of 'runs' to mark discontinuities in data
Exclude data resolution less than 5.999 Angstrom or greater than 2.800 Angstrom
MTZ out d44n01 d44n01-highres_scala1.mtz Browse View
Convert to SFs & Wilson Plot
Estimated number of residues in the asymmetric unit 415
Use dataset name as identifier to append to column labels
Include the intensities in the output MTZ file(s)
Data Harvesting
Create harvest file in project harvesting directory
Define Output Datasets
The input file contains a single dataset, which will be transferred to the output file.
Crystal d44n01 belonging to Project d44n01
Dataset name highres
Scaling Protocol
Scale on rotation axis with secondary beam correction with isotropic Bfactor scaling
Define scale ranges along rotation axis by rotation interval 5
Secondary beam correction maximum number of spherical harmonics 6
Independent Bfactors defined by rotation interval 20
Run
Save or Restore
Close

Figure 9. The *Scale and Merge Intensities* task window in CCP4i. Mandatory fields are highlighted in color.

- The scaling job is started by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.

- When the job is finished, examine the scaling statistics by selecting *View Files from Job…View Log File* from the administration pane. In the log file window, select *Show Summary*.
    - The overall R_{merge} should be very low, typically ≈0.05 for an excellent data set. Overall R_{merge} values > 0.10 are cause for concern. For a typical data set the R_{merge} values by shell should increase monotonically from low- to high-resolution shells. You may consider disregarding high-resolution data with R_{merge} > 0.40, and re-scaling with reduced resolution limits.
    - The overall I/σ(I) value should typically be ≈20. An overall I/σ(I) < 10 is cause for concern. For a typical data set I/σ(I) should decrease monotonically from low- to high-resolution shells. You should disregard shells with I/σ(I) < 2, and re-scale with reduced resolution limits.
    - Examine the completeness of the data set. Data that is 85-90% complete should be sufficient to solve a structure, although more completeness is better if practical. A quality data set will also have approximately the same degree of completeness in each shell, with perhaps a monotonic fall-off at high resolution where spot intensities are weaker or are limited to the corners of the images. Gaps in completeness in low- or mid-resolution shells may indicate problems with ice rings and/or integration.
    - Examine the multiplicity of the data set. A typical data set will have an average multiplicity of ≈4. This is a measure of the average number of times a reflection intensity I_{hkl} (or its Friedel mate, $I_{-(hkl)}$) has been independently measured. Higher multiplicities will result in a more precise data set. There may be a fall-off in multiplicity at high resolution where spot intensities are weaker.

Merging multiple data sets in CCP4i

Frequently in protein X-ray crystallography it is necessary to combine several datasets in order to solve a structure. Such situations might include:

- combining several datasets at from different phi rotations of the same crystal. This situation might arise from an interrupted data collection run where the initial data set was not sufficiently complete, and for which it was impossible or impractical to resume the run exactly where it left off. Combining two sets will allow the construction of a suitably complete data set
- combining datasets from the same crystal using different camera distances. This situation is very useful when a crystal has large unit cell dimensions (and therefore closely spaced spots), where it is difficult to collect a complete dataset which includes high-resolution data as well as well-resolved low-resolution reflections. In this case the low resolution data can be collected as a separate dataset with a longer camera distance, allowing better separation of low-resolution reflections. Overlapping low-resolution reflections are discarded in the high-resolution data set.
- combining several datasets from different crystals of the same protein in the same space group. This situation might arise when crystals have a limited lifetime in the X-ray beam, and no single data set is complete enough for structure solution.

To merge datasets, the second and subsequent datasets must be renumbered so that batches of reflections (collections of reflections from a frame of data) will have unique, non-conflicting batch numbers. The resulting sorted datasets are then combined and sorted by reflection, and then finally re-scaled to render them consistent with each other. The following procedure is typical for merging and scaling two data sets:

- Open a CCP4i session as previously described previously.
- Select the *Data Reduction* module in CCP4i (upper left menu bar) and click on *Sort/Modify/Combine MTZ files*. A task window will open (Figure 10).

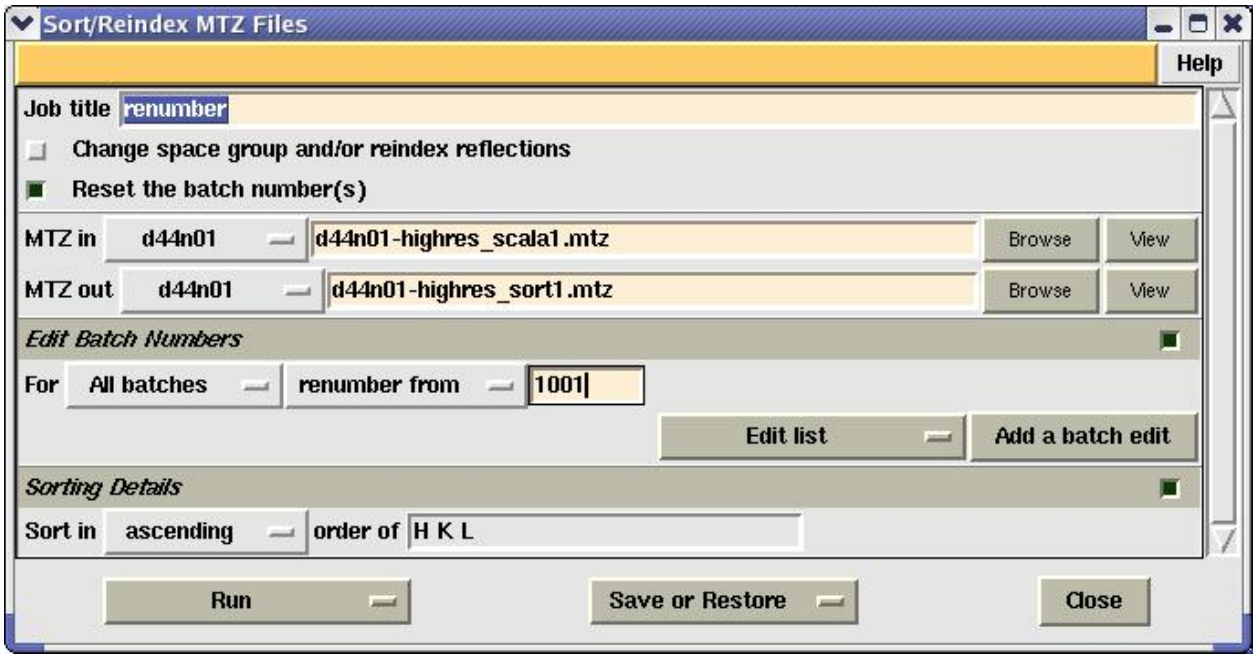


Figure 10. *Sort/Modify/Combine MTZ files* task window. Mandatory fields are highlighted in color.

- Enter a job name (*e.g.*, `renumber`), and select the appropriate MTZ input filename. An output file name will be generated, or you can change it to something else.
- Select *Reset the Batch number(s)* and enter a number for the first batch. This number should be larger than the highest batch (frame) number in the batch of the other dataset. It is simplest to add a multiple of 1000 to the original batch number.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.
- Open a new *Sort/Modify/Combine MTZ files* task window. Enter a job name (*e.g.*, `combine`) and enter the MTZ filename of the renumbered MTZ file from the previous steps. Click on *Add File* and enter the MTZ filename of the scaled intensities corresponding to the dataset you wish to combine it with. Finally, select an output MTZ filename.

- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.
- To scale and merge the sorted files, open a *Scale and Merge Intensities* task window.
- Enter a job name (*e.g.*, `merge`) and select as your input MTZ file the sorted and combined MTZ file created in the previous job. Select and output MTZ filename, and in the *Define Output Datasets* section check *Combine all input datasets into a single output dataset*. Change the output dataset name to something descriptive like `all`. Optionally, enter the estimated number of residues per asymmetric unit to get accurate Wilson Plot statistics, including the average estimated *b*-factor for the dataset.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly. Examine the scaling statisitics to verify that the combined data set is satisfactory. Combined datasets may not have monotonically varying values of R_{merge}, I/σ(I), or multiplicity by shell because of discontinuities in the merged data. However, the merged data should still have overall statistics that conform to what is expected for a usable dataset.
- The output MTZ file from this procedure is ready for further processing as described in the next section, Model Building and Refinement.

Reindexing Data Sets in CCP4i

Sometimes a space group is generally known but the exact space group including screw axes is not immediately known, and the data set must be re-indexed to conform to standard conventions later. For example, you may know that a particular crystal is in the primitive orthorhombic space group (*e.g.*, P222, $P2_12_12_1$, $P2_12_12$, $P2_122_1$, $P22_12_1$, $P222_1$, $P22_12$, $P2_122$). Of these space groups, only P222, $P2_12_12_1$, $P2_12_12$, and $P222_1$ are recognized as standard space groups. The others are non-standard variants in which the h, k, l indices have been permuted. To convert one of these non-standard space groups into a standard one, the reflection data indices must be appropriately swapped. For example to convert reflection data from $P22_12_1$ to the standard $P2_12_12$, it is necessary to rearrange the indices *hkl* into *klh*. This is conveniently done in CCP4i:

- Open a CCP4i session as previously described previously.
- Select the *Reflection Data Utilities* module in CCP4i (upper left menu bar) and click on *Reindex Reflections*. A task window will open (Figure 11).
- Enter a job name (*e.g.*, `reindex`), and select the appropriate MTZ input filename. An output file name will be generated, or you can change it to something else.
- Under the *Reindex Details* section of the form, select *entering reflection transformation*. In this example, we have selected *h=k, k=l, l=h* to permute the indices *hkl* to *klh*.

- Check the box *Change spacegroup to and e*nter the proper, standard space group, here *P21212*.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.

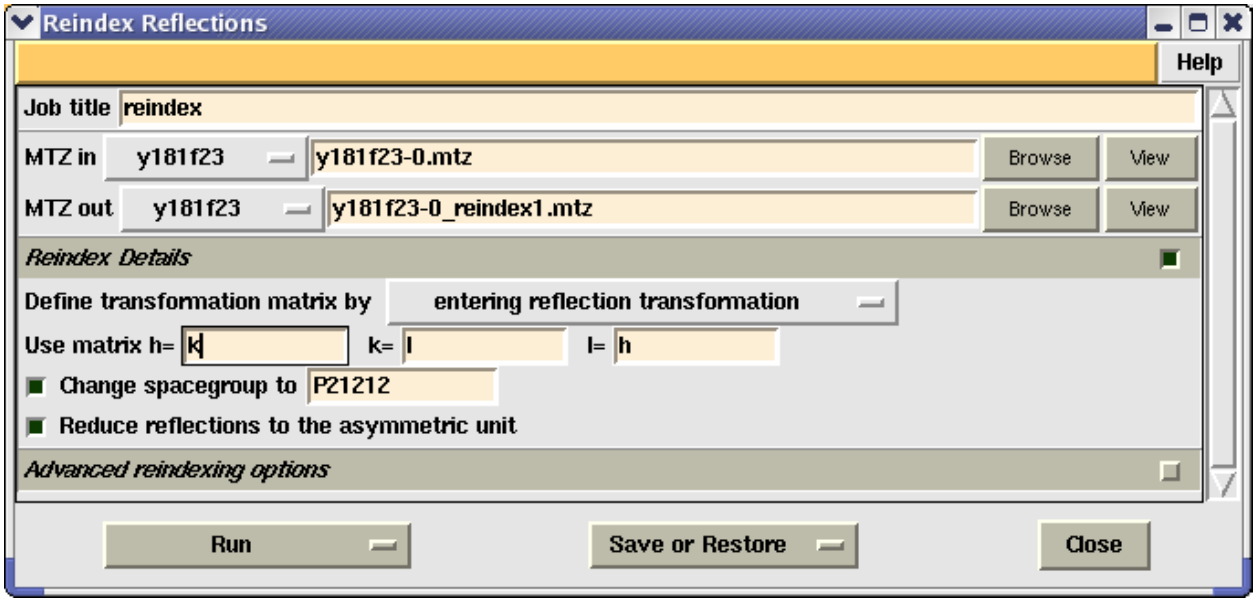


Figure 11. *Reindex Reflections* task window. Mandatory fields are highlighted in color.

Additional Resources

- HKL Research, Inc.: http://www.hkl-xray.com/
- CCP4: http://www.ccp4.ac.uk/

Model Building and Refinement

Obtaining a structure from your data is an iterative process that requires supplying a model of your protein structure, comparing it to the electron density map derived from the data (and model), and rebuilding the model. When the model and electron density map are in sufficient agreement, the model is regarded as a good approximation of the protein structure.

The most difficult problem in this modeling process is obtaining information about the *phase* of the observed reflections. (The intensities are accurately measured in your experimental data set.) In order to produce accurate electron density maps, it is essential to have both accurate intensity and phase information. Approximate phases can be obtained by collecting additional data on heavy atom derivatives of the same protein (multiple isomorphous replacement), by examining anomalous scattering of endogenous heavy atoms in the protein (useful for certain metalloenzymes or selenomethionine-substituted proteins), or by using a starting model derived from a homologous protein (molecular replacement). This edition of the handbook will only treat the latter case, molecular replacement.

There are many software suites available for the analysis, modeling, and refinement of protein X-ray diffraction data. The methods described here utilize the *CNS* suite (Crystallography and NMR System) and/or the *CCP4* (Collaborative Crystallography Project) suite, and some other ancillary programs such as *EPMR* and *Phaser*. Model building and examination of electron density maps is done using Alwyn Jones' program *O* or Paul Emsley's *Coot*.

Preparing SCALEPACK data for analysis

Converting scalepack data to CNS format. Prior to refinement in CNS, it is helpful to do a series of file conversions. The first of these is to convert the SCALEPACK reflection file into the *.mtz* format that CCP4 recognizes. If you processed your data in MOSFLM and SCALA, this step is not necessary. The Linux script `sca2mtz` (File 4) will do this. To run this script the CCP4 suite must be enabled by running the appropriate source command which is typically aliased to the command `ccp4setup`. The input file (hklin), the output file (hklout), the space group (symmetry), and the log filename should be edited as required. To run the command type `sca2mtz` at the prompt. Examine the log file to ensure that the program ran satisfactorily before proceeding.

File 4
`sca2mtz`

```
scalepack2mtz hklin hica15.sca hklout hica15a.mtz >sca2mtz.log << eof-scale
ANOMALOUS NO
SYMMETRY C2
END
eof-scale
```

Truncation of reflection data. The Linux script `trunc` (File 5) calls the CCP4 program `truncate` which converts the intensity data (in `.mtz` format) output by `sca2mtz` to

structure factors. In addition, `truncate` also provides useful information about the data set, including the average *b*-factor for the data set (from a Wilson plot), the approximate percentage of the crystal lattice occupied by protein, and the scattering anisotropy of the crystal, along with other information. (If you have processed your data in MOSFLM and SCALA as described in the previous section, your data has already been converted to structure factors by `truncate`, and this step is unnecessary, and you can proceed directly to molecular replacement and/or model-building and refinement.) The input file (hklin), the output file (hklout), the number of residues in the asymmetric unit (nresidues) and the log filename should be edited as required. To run the script type `trunc` at the prompt. To run this script the CCP4 suite must be enabled by running the appropriate source command which is typically aliased to the command `ccp4setup`. Examine the log file to ensure that the program ran satisfactorily before proceeding.

File 5
`trunc`

```
truncate hklin hica15a.mtz hklout hica15b.mtz > trunc.log <<eof-truncate
TRUNCATE YES
NRESIDUES 1374
END
eof-truncate
```

Converting MTZ files to CNS format

The final step before beginning data analysis in CNS is to convert the reflection file data of a CNS-compatible format. (This will not be necessary if using CCP4 to do structure refinement.) The Linux script mtz2fobs (File 6) does this conversion. The input file (hklin), the output file (hklout), and the log filename should be edited as required. To run the script type `trunc` at the prompt. To run this script the CCP4 suite must be enabled by running the appropriate source command which is typically aliased to the command `ccp4setup`. Examine the log file to ensure that the program ran satisfactorily before proceeding. The `.fobs` file is the starting point for data analysis.

File 6
`mtz2fobs`

```
mtz2various hklin hica15b.mtz hklout hica15.fobs > mtz2fobs.log <<eof-various
OUTPUT XPLOR
LABIN F=FP SIGF=SIGFP
END
eof-various
```

This task can also be carried out in the CCP4i interface by the following steps:

- In the CCP4i main task window select *Reflection Data Utilities* from the task menu and click on *Convert from MTZ*. A task window will open (Figure 11).
- Enter a job name (*e.g.*, `mtz2fobs`) and the input file name (the sorted structure factor MTZ file for the entire dataset)

- A list of available fields will appear in the *MTZ File Labels* section. The only fields typically required for structure solution are *FP* and *Sigma*, which correspond to the structure factor and its standard error. Set all other fields to *Unassigned* using the drop-down menus.

- Select an output file name. It is suggested that you make the file extension `.fobs` to be consistent with the instructions in this methods manual.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.

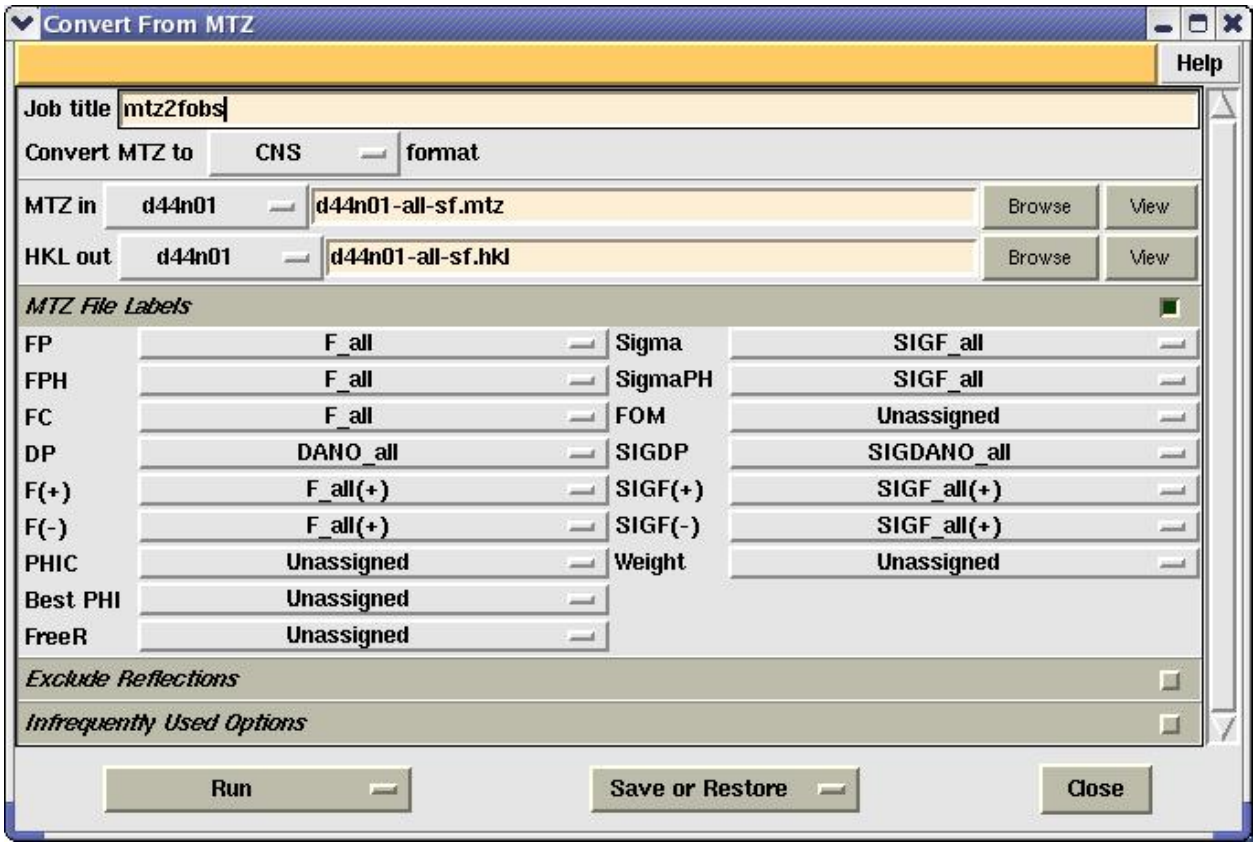


Figure 11. *Convert from MTZ* task window. Required fields are highlighted in color. Available data fields in MTZ file are listed in the MTZ File Labels section.

- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.
- The resulting `.fobs` file is now ready for use in structure solution.

Obtaining initial phases by molecular replacement

The simplest method of obtaining phase estimates for X-ray diffraction data analysis is *molecular replacement*, which involves building a provisional model of the target protein based on the structure of a highly homologous protein, and placing it in the appropriate orientation in the unit cell. The initial phases are calculated based on the positions of all the atoms in the

molecular replacement model, and such phases are often sufficient to obtain a usable electron density map that can be used to refine the structure of the target protein. Two excellent tools for solving structures by molecular replacement are *EPMR* and *Phaser*, both of which are detailed here.

Constructing a molecular replacement model

For any molecular replacement solution, it is necessary to construct a reasonable molecular replacement search model. Select a molecular replacement protein that is as homologous as possible to the target protein, and examine a sequence alignment of the two proteins. A molecular solution replacement may be possible if the proteins are more than 30% identical. The molecular replacement protein should be modified as follows to make it a similar as possible to the target protein:

- If the molecular replacement protein has extra residues, either internally or at the N- or C-termini, remove them.
- Leave as is any residues that are identical in both proteins
- For mismatches, change the molecular replacement residue to Ala except:
    - Pro, Gly or Ala residues in molecular replacement model should be left as is
    - Gly should be used where Gly appears in the target protein
    - No substitution is necessary for Asn/Asp or Gln/Glu
    - Phe in the molecular replacement protein is allowed to subsitute for Tyr in the target protein
    - Val in the molecular replacement protein is allowed to subsitute for Ile in the target protein

The necessary modifications can be easily made using O, Coot, or Swiss-PDB Viewer. (Note: for solving the structure of mutant proteins, the ideal search model is an existing solved structure of the wild-type protein. No modifications need be made to the residues of the molecular replacement model in this case.) For the purpose of generating an initial electron density map it is probably wise to remove all cofactors (*e.g.*, coenzymes, metal ions), bound species (*e.g.*, buffers, solvents, ions), and solvent.

Finding molecular replacment solutions using EPMR

Before an electron density map can be generated, it is necessary to place the search model (molecular replacement protein) in the appropriate location of the unit cell. There are a number of programs capable of doing this, but among the best is EPMR, the instructions for which are described here. The first task is to convert the truncated `.mtz` file output by `truncate` to a format readable by EPMR. The unix script `mtz2epmr` (File 7) accomplishes this task. The input (hklin), output (hklout) and log files should be edited as required. This same task can also be accomplished in the CCP4i environment by choosing the task *Convert from MTZ* in the *Reflection Data Utilities* menu. The CCP4i task window for carrying out the actions of File 7 are shown in Figure 12.

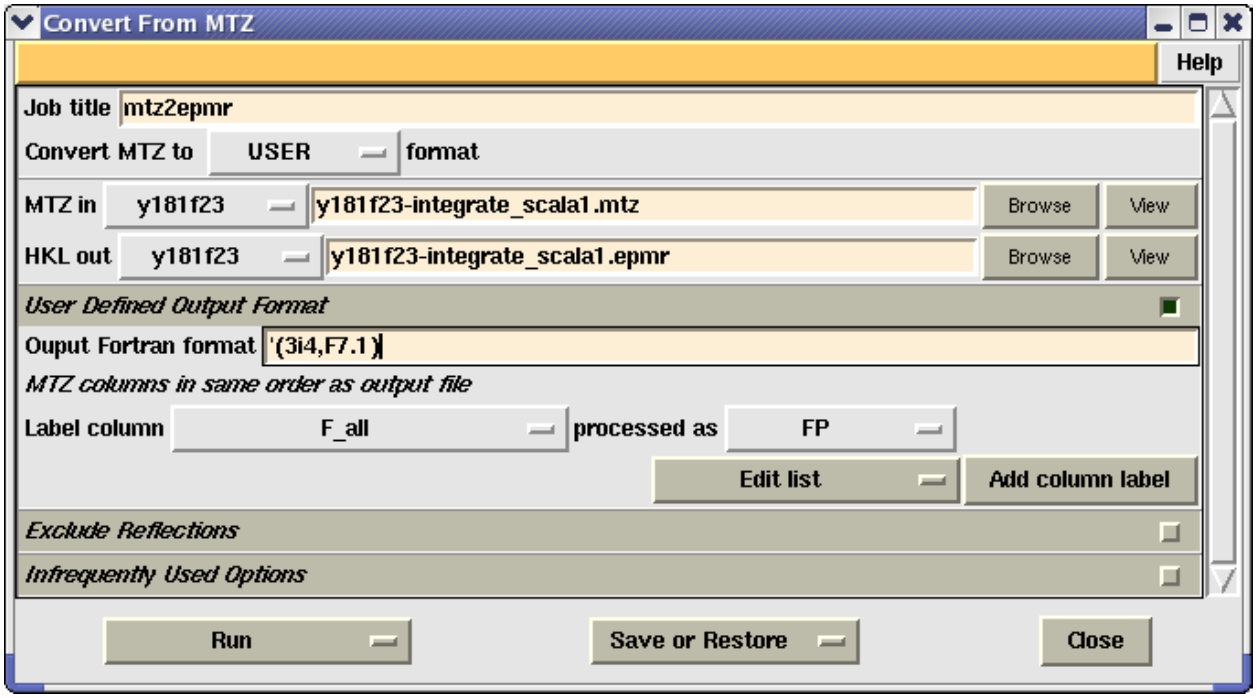


Figure 12. *Convert from MTZ* task window. Required fields are highlighted in color. Data fields in MTZ file that are to be converted to user-defined format are listed in the MTZ File Labels section.

File 7
`mtz2epmr`

```
mtz2various hklin hica15b.mtz hklout hica15.epmr > mtz2epmr.log <<eof-various
LABIN F=FP
OUTPUT USER '(3I4,F7.1)'
END
eof-various
```

EPMR also requires an additional file that contains information about the unit cell dimensions and the space group number. This file should contain a single line in the format in which the values of *a*, *b*, *c*, α, β, γ, and the *International Tables* space group number are entered separated by spaces. The unit cell parameters and space group number can be found in the log file of `truncate`. Give the file the .cel extension. File 8 is an example for a *C2* crystal (space group #5):

File 8
epmr `.cel` file

```
232.66 144.73 52.41 90 93.96 90 5
```

Running EPMR. EPMR uses an efficient evolutionary search algorithm to find one of many good fits of the search model to the reflection data during each trial. The search is repeated for many trials, starting with different initial orientations of the search model. The results of the best of these trials is assumed to be (and often is) close to global best fit, providing a good model for estimating phase data and constructing the first electron density map. The program is customizable by including various switches in the command line, some of which are outlined below:

- `-o` *filename* sets the stem for the filenames of the output PDB files, which will look something like *filename*`.1.best.pdb`.
- `-m`*n* instructs EPMR to place *n* molecules of the search model into the unit cell. The default is to place one molecule in the unit cell
- `-t`*n* instructs EPMR to use the correlation coefficient *n* as the cutoff value for determining what is a satisfactory molecular replacement solution. When placing more than one molecule in the unit cell, it is usually desirable to set this value to 1.0 to force an more exhaustive search for the best fit for the first molecule placed. This often improves the chance of success for finding a satisfactory solution for multiple placements. The default is a correlation coefficient of 0.45 for one molecule or 0.30 for the first of multiple molecules.
- `-h`*n* gives EPMR the high-resolution limit of data to be used in the search. The default value is 4 Å. Occasionally, using slightly higher resolution data can help find a satisfactory solution. This value should normally be set to 5Å or higher resolution.
- `-l`*n* gives EPMR the low-resolution limit of data to be used in the search. The default value is 15 Å. If accurately measured low resolution reflections are available, including data out to 25-30Å can be useful.

The general format for invoking the program is:

```
epmr –o filestem filename.cel filename.pdb filename.epmr
```

where `filestem` is the stem of the output PDB filename, `filename.cel` is the unit cell information file, `filename.pdb` is the molecular replacement search model in PDB format, and `filename.epmr` is the reflection list file in EPMR format. The command line, which can be quite long, is best put into an executable Linux script file named `epmr.sh`, an example of which is shown in File 9. The command can be invoked to run in the background by typing `epmr.sh &` at the prompt.

File 9
A typical EPMR executable file

```
epmr –m3 –t1.0 –o 3dimer hica08.cel dimer.pdb hica08.epmr > 3dimer.log
```

The script in File 9 will do an exhaustive search (correlation coefficient of 1.0) to place 3 molecules of `dimer.pdb` in the unit cell described by `hica08.cel`, using `hica08.epmr` reflection data. The best fits for the three placed dimers will be written out as

`3dimer.1.best.pdb`, `3dimer.2.best.pdb`, and `3dimer.3.best.pdb`. The real-time output of the program will be sent to the file `3dimer.log`, which can be monitored by using the `tail -f` command. EPMR, even as efficient as it is, will take a substantial amount of time to find a molecular replacement solution for a large unit cell, especially if multiple molecules must be placed. It is best run as an overnight job.

Preliminary determination of suitability of the molecular replacement search. A decent molecular replacement solution will have an *R*-factor no larger than ≈0.45. If *R*>0.50 it is unlikely that the molecular replacement solution will be useful. If the *R*-factor is satisfactory, then the molecules placed in the unit cell by EPMR should be examined for overlaps with themselves or with symmetry-generated partners by loading them into Swiss PDB Viewer or O. (The operation of O is described later.) If there are no obvious overlaps, and the symmetry-generated molecules pack well into the unit cell, you should proceed, else you should re-evaluate your molecular replacement solution and perhaps try again using different conditions.

Preparing EPMR output for use by CNS. If you have placed several molecules of a search model into the unit cell, they should be consolidated and reformatted before proceeding. First, the files should be concatenated using a text editor; any remark files can be removed. Next, the file should be reformatted so that each protein chain has a different SEGID. This is most conveniently accomplished in MOLEMAN2. To invoke the program, type `moleman2` at the prompt. To read in the PDB file, type `RE` *filename*`.pdb` where *filename*`.pdb` is the concatenated PDB file. To rename the segid's, type `CH AS` and name the segid's A, B, C, D, etc. At this point it is also useful to set the *b*-factors to the value estimated by `truncate`. To change all the B-factors, type BF LI and enter the *b*-factor value from `truncate` as both the high and low limit. To save the changes to the file type `WR newfile.pdb`, where `newfile.pdb` is the new filename of the modified PDB file. MOLEMAN2 is a very powerful program for modifying PDB files, and is well worth learning more about.

Finding molecular replacement solutions using Phaser

Phaser is another powerful molecular replacement program that is now integrated into the latest release of CCP4. Data that cannot be solved by EPMR can often be solved by *Phaser*, and *vice versa*. Phaser is most conveniently run via CCP4i, and one feature of the CCP4i task window can be used to estimate the number of protein molecules present in the asymmetric unit prior to running either Phaser or EPMR.

Estimating the number of protein molecules in the asymmetric unit. A utility within Phaser can utilize Matthews Probability calculations to estimate the most likely number of protein molecules within the asymmetric unit of the unit cell.

This task can also be carried out in the CCP4i interface by the following steps:

- In the CCP4i main task window select *Molecular Replacement* from the task menu and click on *Phaser*. A task window will open (Figure 13).
- Enter a job name (*e.g.*, `matthews`) and the input file name (the sorted structure factor MTZ file for the entire dataset)

- Under *Mode for molecular replacement*, select *cell content analysis*.
- Under *Composition of the asymmetric unit*, choose *protein* and enter the molecular weight of the search model, and the number of these molecules you expect in the asymmetric unit. If you don't know how many search models are reasonable to enter, try "1."
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.
-

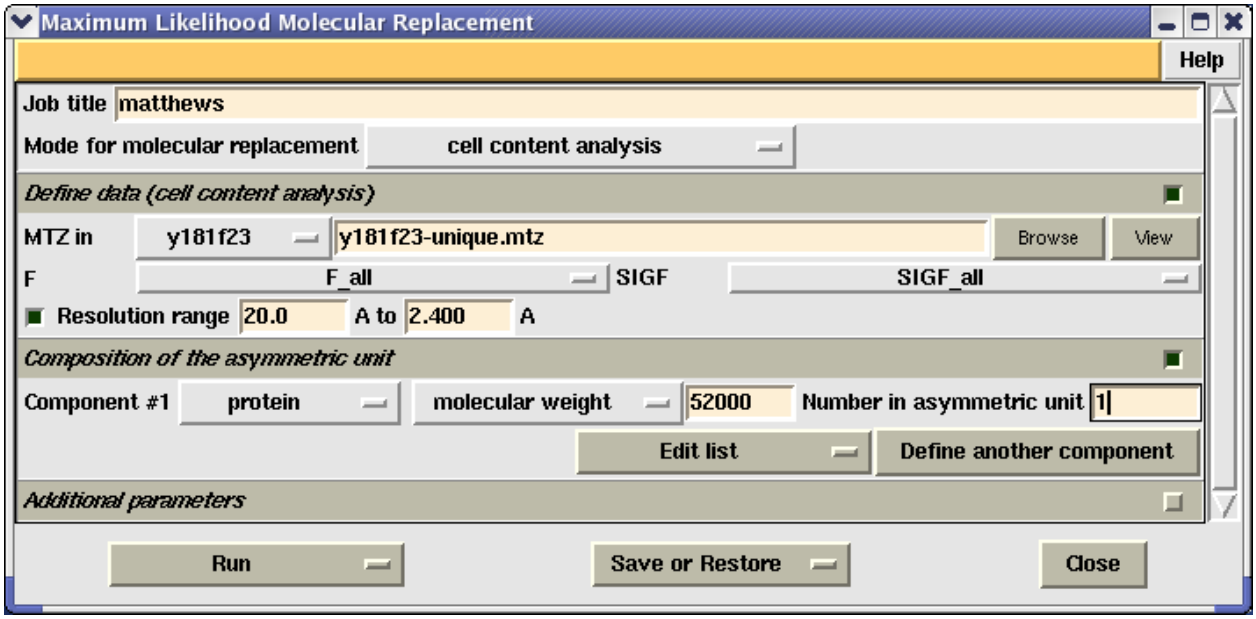


Figure 13. *Phaser* task window set up for Matthews probability estimation. Required fields are highlighted in color.

Performing molecular replacement calculations in Phaser. Phaser is a fast, highly automated program for finding molecular replacement solutions for multiple protein molecules (search models) in an asymmetric unit. Phaser is conveniently run in the CCP4i environment:

- In the CCP4i main task window select *Molecular Replacement* from the task menu and click on *Phaser*. A task window will open (Figure 14).
- Enter a job name (*e.g.*, `phaser`) and the input file name (the sorted structure factor MTZ file for the entire dataset)
- Under *Mode for molecular replacement*, select *automated search*.
- Under *Composition of the asymmetric unit*, choose *protein* and enter the molecular weight of the search model, and the number of these molecules you expect in the asymmetric unit based on Matthews probability analysis.

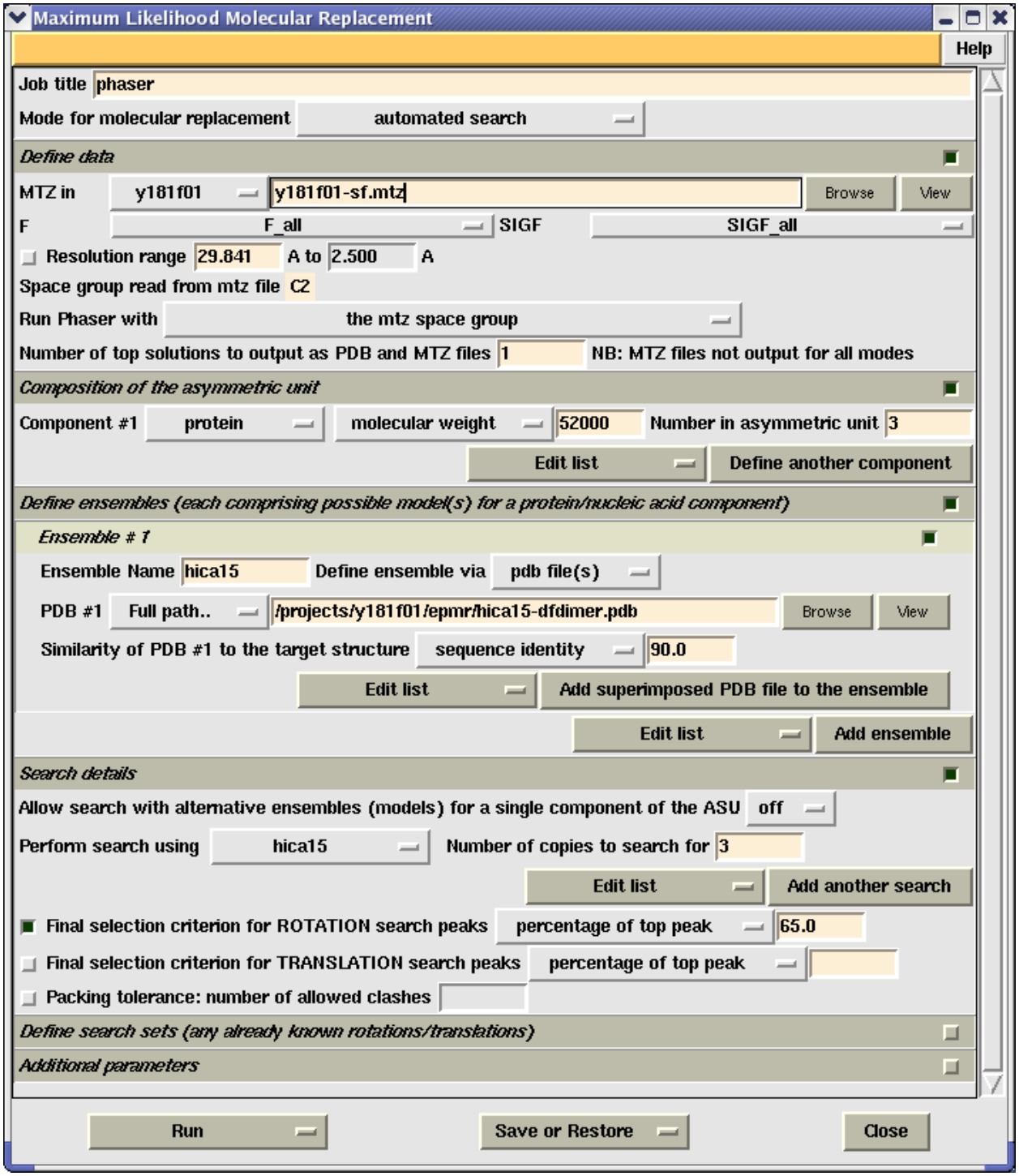

Maximum Likelihood Molecular Replacement
Help
Job title phaser
Mode for molecular replacement automated search
Define data
MTZ in y181f01 y181f01-sf.mtz Browse View
F F_all SIGF SIGF_all
Resolution range 29.841 A to 2.500 A
Space group read from mtz file C2
Run Phaser with the mtz space group
Number of top solutions to output as PDB and MTZ files 1 NB: MTZ files not output for all modes
Composition of the asymmetric unit
Component #1 protein molecular weight 52000 Number in asymmetric unit 3
Edit list Define another component
Define ensembles (each comprising possible model(s) for a protein/nucleic acid component)
Ensemble # 1
Ensemble Name hica15 Define ensemble via pdb file(s)
PDB #1 Full path.. /projects/y181f01/epmr/hica15-dfdimer.pdb Browse View
Similarity of PDB #1 to the target structure sequence identity 90.0
Edit list Add superimposed PDB file to the ensemble
Edit list Add ensemble
Search details
Allow search with alternative ensembles (models) for a single component of the ASU off
Perform search using hica15 Number of copies to search for 3
Edit list Add another search
Final selection criterion for ROTATION search peaks percentage of top peak 65.0
Final selection criterion for TRANSLATION search peaks percentage of top peak
Packing tolerance: number of allowed clashes
Define search sets (any already known rotations/translations)
Additional parameters
Run
Save or Restore
Close

Figure 14. *Phaser* task window set up for molecular replacement solution. Required fields are highlighted in color.

- In the *Define ensembles* section, provide an ensemble name and enter the filename of the search model, as well as an estimate of the homology of the search model to the protein of interest. Do not enter 100% if the two protein are not perfectly identical. Underestimates are better than overestimates of homology.

When using a wild-type protein as a search model for site-directed mutants, a value of 90% is OK.

- Under *Search Details*, enter the ensemble name and the number of copies to be placed in the asymmetric unit.
- To enhance the probability of finding an appropriate solution, it is recommended that you check the box *Final selection for rotation search peaks* and enter *65* for *percentage of top peak*. This will retain more possible solutions from cycle to cycle. (The default is 75%.)
- It is also suggested that you increase the *Packing tolerance* by checking the appropriate box and allowing for 10-20 clashes. This will slow the solution somewhat, but will prevent the elimination of solutions that have molecular clashes between mobile and disordered portions of the search models when packed in the unit cell.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status. *Phaser* jobs can take 2-24 hours, depending on the complexity of the problem and the various selection criteria. CCP4i jobs will continue to run even if you exit CCP4i and logout of your account.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.

Preparing the first electron density map

Preparing the first map and doing the subsequent refinements heavily uses CNS and the molecular display program O (described later). Alternatively, you can use the CCP4 program Refmac for refinement and the molecular display program Coot. Beginners will probably find the web interface of CNS the simplest to use when setting up CNS script files, and CCP4i easiest for running CCP4 tasks. More experienced users are more likely to modify previously written script files to run CNS or CCP4 tasks. While both approaches are discussed here, the CCP4/Coot packages are somewhat easier to manage and are better integrated, and are recommended.

Preparing the first map is an exciting and expectant time. You are either rewarded with immense joy of actually seeing clear electron density delineating the path of the main chain and positions of many side chains, or you suffer the crushing disappointment of hash. Either way, here is how you generate the first map.

Initial model refinement using CNS

Running CNS scripts. The general method of running scripts is to type the command `cns_solve<filename.inp>filename.log` at the prompt. This will run the script `filename.inp` and output a log file to `filename.log`. You should always examine the log file to ensure that the program completed successfully. Before you run CNS the first time it is usually necessary to enable the CNS modules by running the appropriate source command, which is normally aliased to the command `cnssetup`.

Constructing a cross-validated reflection file. This is an important step in your structure determination. You are going to set aside a portion of your reflection data as a *test* data set that will not be used in the construction of the model, but will be used to independently measure how well your model fits the raw data. The nature of the iterative procedure of modeling and refining biases the electron density map to conform to the model, no matter how wrong it may be. The set-aside test data is your guard against falling too deeply into this trap. Modify the file `make_cv.inp` so that it will set aside 5-10% of your reflections as test reflections. If your data is nearly complete (>95%) you should use the maximum value (10%). The input file should be the `.fobs` file you created previously. The output file should be given the `.cv` extension to uniquely identify it. This `.cv` file will be used to do all subsequent refinements.

Generating CNS topology files. CNS requires, in addition to a PDB file, a molecular topology (`.mtf`) file that contains information about molecular connectivity and geometrical constraints necessary to guide the refinement. You must generate a new `.mtf` file whenever you have add or delete atoms from your model. It is typical to generate a new `.mtf` file at the beginning of each refinement cycle. To generate an .mtf file, edit and run the script `generate_easy.inp`. The required input is a `.pdb` file, and the outputs are a (new) `.pdb` file and an associated `.mtf` file. Be sure you have included the necessary topology (.top) and parameter (.par) files for ions, water, and hetero-compounds included in your model. Hetero compound `.pdb`, `.top`, and `.par` files not included with the CNS distribution can usually be downloaded from the HIC-UP server.

Fine-tuning the molecular replacement solution. The molecular replacement solution should be fine-tuned by adjusting the position of the model, including the subunits independently of each other, via a rigid-body refinement. This is accomplished by the CNS module `rigid.inp`. Required inputs for `rigid.inp` are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically 15.0–4.0 Å), the segid names to be minimized (typically all of them), and the name of the output file (typically `rigid.pdb`). The highest resolution shell should not be set too low a value else the refinement may not be able to move the model far enough to find the global mininum best fit. Rigid body refinement needs only be run this one time. It does not have to be run again during the refinement procedure. Normally, the *R*-factor will decrease by 5% or more during rigid-body refinement, and this is usually a good sign that things are going well. You should also monitor your *R-test* vs. your *R-free* values at this point and after all subsequent refinement steps. *R-test* is the residual experimental data used for refinement not explained by your model; *R-free* is the residual test data (the 10% you put aside in your `.cv` file) that is not explained by your data. Normally *R-test* is lower than *R-free* (presumably because of slight model bias) but generally these values are no more than 5% (0.05) apart. If *R-free*–*R-test* > 0.05 you should investigate further or take steps to reduce model bias, such as simulated annealing.

Calculating electron density maps. You should calculate two types of electron density maps for modeling purposes. A $2F_o–F_c$ map is calculated from 2 × the observed minus the calculated electron density. This map resembles the electron density of the target molecule and should largely define the main and side chains of the model. A $F_o–F_c$ map is calculated from the

difference of the observed and calculated electron densities. This map is useful for identifying region of electron density not explained by the model. Positive F_o–F_c density indicates that there is electron density present not explained by the model, *e.g.*, a missing cofactor, solvent, buffer, or ion molecule, or a misplaced (or missing) main- or side-chain; Negative F_o–F_c density indicates that there is less electron density in the data than is predicted by the model, *e.g.*, a misplaced main- or side-chain. Electron density maps are generated by the CNS module `model_map.inp`. Experienced users usually save two separate CNS scripts named `2fofc.inp` and `fofc.inp` to save time generating maps. Required inputs are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically all of the data), the type of map (`u=2` for $2F_o$–F_c maps and `u=1` for F_o–F_c maps) and the name of the output map file (typically `2fofc.map` or `fofc.map`).

The electron density maps are very large, and should be immediately converted into O-maps, which are about $1/10^{th}$ the size of the CNS maps. After the conversion, the large and now unnecessary `.map` files should be deleted. To do this, type the command `map_to_omap *.map` at the prompt. This will invoke the utility MAPMAN and do the necessary conversions. If the conversion fails due to lack of memory (common for large maps) you will need to set the environment variable MAPSIZE to a larger value. Choose a value slightly larger than that shown in the MAPMAN error message, *e.g.* `setenv MAPSIZE 10000000`. Note that the environment variable MAPSIZE is all capitalized.

The resulting model and maps should be examined in O (described later) to see if they are usable. Examine the display to see if much of the model is contained within the $2F_o$–F_c map. In addition, examine the F_o–F_c map for key areas of positive density, e.g., known metal ions, cofactors, or bound inhibitors or substrates. If you observe positive density in the correct areas of the model, chances are very good your molecular replacement solution is usable, and you should proceed with refinement.

Further model refinement using CNS

After the initial rigid body refinement, the model is typically subjected to simulating annealing, which essentially "shakes up" the model in a random way, followed by slow "cooling" to find a better, less model-biased fit to the experimental data. After this step, the model is typically taken through repeated rounds of whole-molecule minimization, *b*-factor refinement, and the generation of new electron density maps for visualization in O. At the end of each refinement round manual adjustments are made to the main- and side-chains to bring them into better compliance with the electron density map.

At the end of the first round of refinement, residues in the molecular replacement model should be changed to the proper side chain in the target molecule, and oriented properly in the electron density. As the refinement proceeds, it may become obvious that some portions of the protein, notably the N- and C-termini, are not visible in the electron density map, and should be removed. Alternatively, as the refinement proceeds, new regions of electron density may become obvious, allowing the addition of residues to the model, especially at the N- and C-termini.

When the *R*-factor has dropped to ≈0.30, it is appropriate to begin adding cofactors and other bound species that are clearly delineated by electron density. Finally, as *R* approaches ≈0.24 or has reached the point that no further improvement is possible, clearly delineated water

molecules can be added. Typical proteins will reach a final *R-free* ≈0.20 or so, depending on the quality of the original data.

The CNS components used in the refinement cycle are described below. Simulating annealing is typically only carried out once. Minimization and *b*-factor refinement are carried out in the order listed after simulated annealing and after each model rebuilding cycle.

Simulated annealing. This procedure "shakes up" the model to remove model bias and then does a whole molecule minimization to find an initial good fit to the experimental data. Simulated annealing is carried out by the module `anneal.inp`. Required inputs are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically all of the data), and the output file (typically `anneal.pdb`). Simulating annealing takes a very long time with large molecules and unit cells, and is best run as an overnight job.

Minimization. Minimization involves taking the input model and fitting it to the reflection data while conforming it to appropriate bond angles and distances for amino acid residues. Minimization is carried out by the CNS module `minimize.inp`. Required inputs are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically all of the data), and the output file (typically `minimize.pdb`).

Group b-*factor minimization.* This process refines the *b*-factors for side chains and main chain atoms as two separate groups. The minimization is carried out by the CNS module `bgroup.inp`. Required inputs are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically all of the data), and the output file (typically `bgroup.pdb`).

Individual b-*factor minimization.* This process refines the *b*-factors for each individual atom separately. The minimization is carried out by the CNS module `bindividual.inp`. Required inputs are the `.pdb`, `.mtf`, and `.cv` files, the unit cell parameters and space group, any extra `.top` or `.par` files required, the resolution range of the reflection data to be used (typically all of the data), and the output file (typically `bindividual.pdb`).

A minimization cycle consists of running `minimize.inp`, `bgroup.inp`, and `bindividual.inp`. in that order. After each cycle, new $2F_o–F_c$ and $F_o–F_c$ maps should be generated from the final output file `bindividual.pdb`. It is a good practice to copy or rename this file to indicate the identity of the crystal and the number of refinement cycles. For example, after the 2nd refinement cycle of the crystal HICA-08, the `.pdb` file might be named `hica08-2.pdb`. A simple Linux script for automating a refinement cycle is shown in File 10. The script files for the CNS modules are arranged so that the output filename for one module is the input filename for the next module. That is, `geneasy.inp` uses `input.pdb` as an input file and outputs `geneasy.pdb` and `geneasy.mtf`; `minimize.inp` uses `geneasy.pdb` and `geneasy.mtf` as input files, and outputs `minimize.pdb`; `bgroup.inp` uses `minimize.pdb` and `geneasy.mtf` as inputs and outputs `bgroup.pdb`, *etc*.

File 10
CNS `refine` script

```
# Roger Rowlett, Feb 2003
# This script takes a PDB file and carries out
# CNS-Minimize, Bgroup, and Bindividual refinement,
# and generates an output PDB file and both 2FoFc
# and FoFc maps based on the refined model.
# *****NOTE*****NOTE*****NOTE*****NOTE*****NOTE*****
# Edit input PDB filename in second line
# Edit output PDB filename in last line
# Run cnssetup prior to execution
# Run ccp4setup prior to execution
# *****NOTE*****NOTE*****NOTE*****NOTE*****NOTE*****
rm geneasy.pdb minimize.pdb bgroup.pdb bindividual.pdb *.map
cp hica01-3c.pdb input.pdb
cns_solve<geneasy.inp|tee geneasy.log
cns_solve<minimize.inp|tee minimize.log
cns_solve<bgroup.inp|tee bgroup.log
cns_solve<bindividual.inp|tee bindividual.log
cns_solve<2fofc.inp|tee 2fofc.log
cns_solve<fofc.inp|tee fofc.log
map_to_omap 2fofc.map
map_to_omap fofc.map
cp bindividual.pdb hica01-4.pdb
```

Refining Structures using CCP4

Constructing a cross-validated reflection file. This is an important step in your structure determination. You are going to set aside a portion of your reflection data as a *test* data set that will not be used in the construction of the model, but will be used to independently measure how well your model fits the raw data. The nature of the iterative procedure of modeling and refining biases the electron density map to conform to the model, no matter how wrong it may be. The set-aside test data is your guard against falling too deeply into this trap. Constructing a cross-validation reflection files can be easily accomplished in the CCP4i environment:

- In the CCP4i main task window select *Reflection Data Utilities* from the task menu and click on *Convert to MTZ and Standardize*. A task window will open (Figure 15).
- Enter a job name (*e.g.*, `set-freer`) and the input file name (the sorted structure factor MTZ file for the entire dataset)
- A filename will be provided for the output data set, or you can change it to something else.
- The crystal, project, and dataset names should be automatically recognized from your input file.
- Select a percentage of the data to set aside for the test (Free-R) set. The default value of 5% (0.05) should normally be adequate.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.

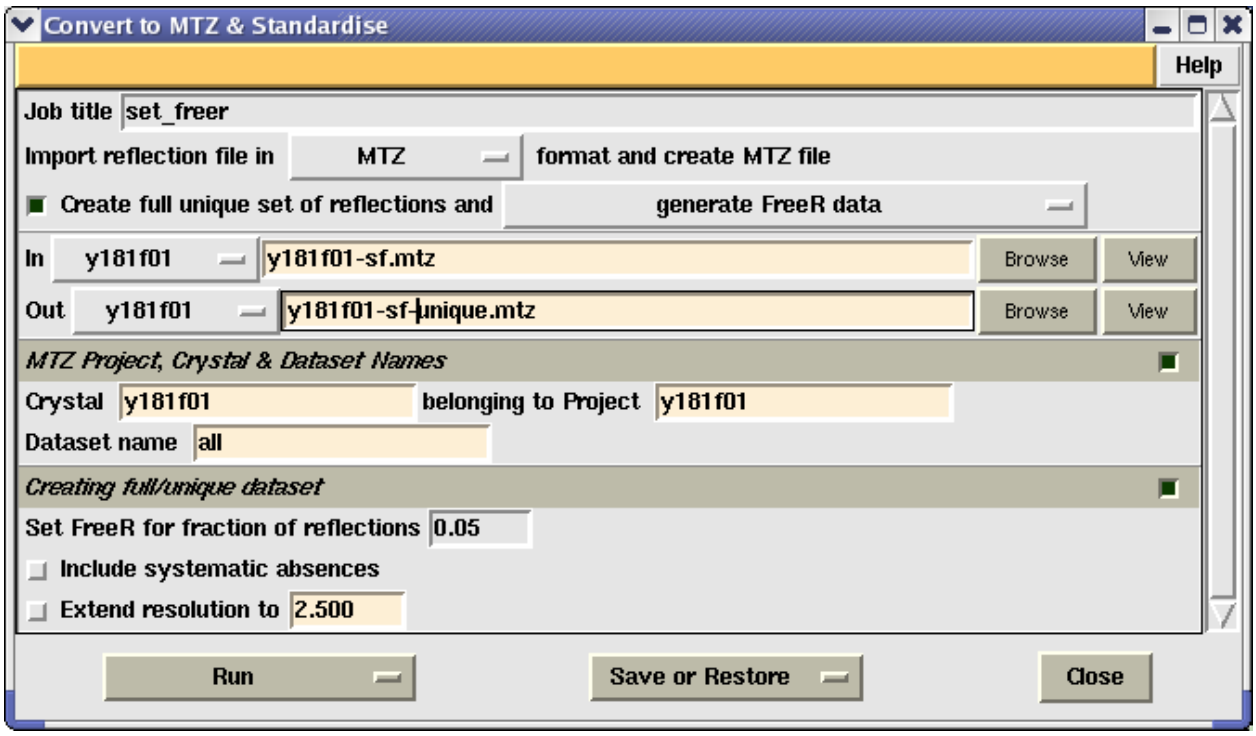

Convert to MTZ & Standardise
Help
Job title set_freer
Import reflection file in MTZ format and create MTZ file
Create full unique set of reflections and generate FreeR data
In y181f01 y181f01-sf.mtz Browse View
Out y181f01 y181f01-sf-unique.mtz Browse View
MTZ Project, Crystal & Dataset Names
Crystal y181f01 belonging to Project y181f01
Dataset name all
Creating full/unique dataset
Set FreeR for fraction of reflections 0.05
Include systematic absences
Extend resolution to 2.500
Run
Save or Restore
Close

Figure 15. Task window for setting the FreeR flag in CCP4 reflection data.

Performing a REFMAC-based refinement. Refmac is a powerful and simple-to-use refinement package. It will perform coordinate and *b*-factor refinement of the model against the structure factor data, and automatically write out phase and intensity information that can be used to construct electron density maps in Coot. Refmac refinement is easily configured in CCP4i:

- In the CCP4i main task window select *Refinement* from the task menu and click on *Run Refmac5*. A task window will open (Figure 16).
- Enter a job name (*e.g.*,`refmac`) and the input file name (the sorted structure factor MTZ file for the entire dataset or the MTZ file from the previous refinement cycle)
- Select *restrained refinement* with no prior *phase information*.
- Filenames will be provided for the output data set and MTZ file, or you can change them to something else.
- The crystal, project, and dataset names should be automatically recognized from your input file.
- The default number of refinement cycles (10) is usually adequate.

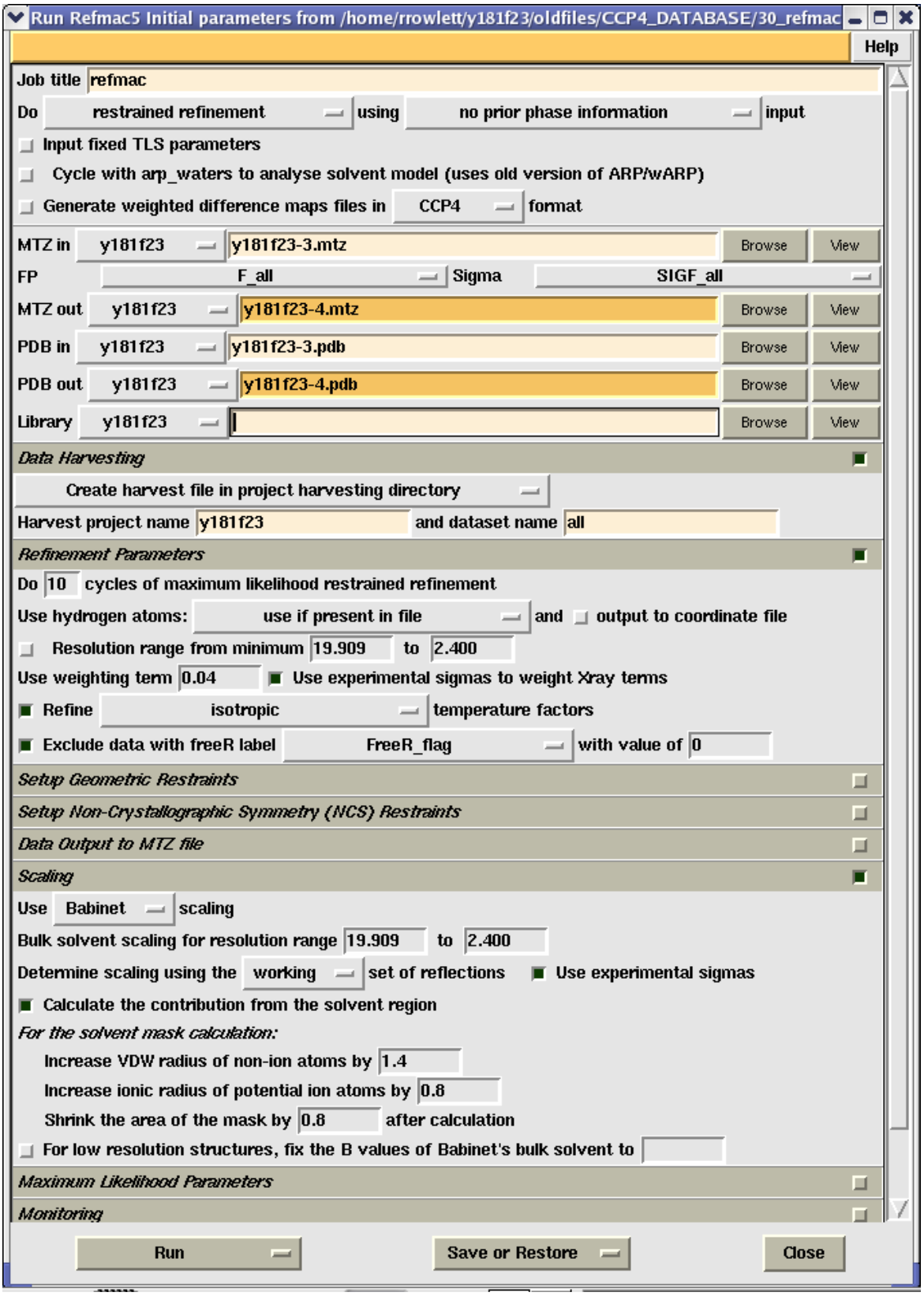

Run Refmac5 Initial parameters from /home/rrowlett/y181f23/oldfiles/CCP4_DATABASE/30_refmac
Help
Job title refmac
Do restrained refinement using no prior phase information input
Input fixed TLS parameters
Cycle with arp_waters to analyse solvent model (uses old version of ARP/wARP)
Generate weighted difference maps files in CCP4 format
MTZ in y181f23 y181f23-3.mtz Browse View
FP F_all Sigma SIGF_all
MTZ out y181f23 y181f23-4.mtz Browse View
PDB in y181f23 y181f23-3.pdb Browse View
PDB out y181f23 y181f23-4.pdb Browse View
Library y181f23 Browse View
Data Harvesting
Create harvest file in project harvesting directory
Harvest project name y181f23 and dataset name all
Refinement Parameters
Do 10 cycles of maximum likelihood restrained refinement
Use hydrogen atoms: use if present in file and output to coordinate file
Resolution range from minimum 19.909 to 2.400
Use weighting term 0.04 Use experimental sigmas to weight Xray terms
Refine isotropic temperature factors
Exclude data with freeR label FreeR_flag with value of 0
Setup Geometric Restraints
Setup Non-Crystallographic Symmetry (NCS) Restraints
Data Output to MTZ file
Scaling
Use Babinet scaling
Bulk solvent scaling for resolution range 19.909 to 2.400
Determine scaling using the working set of reflections Use experimental sigmas
Calculate the contribution from the solvent region
For the solvent mask calculation:
Increase VDW radius of non-ion atoms by 1.4
Increase ionic radius of potential ion atoms by 0.8
Shrink the area of the mask by 0.8 after calculation
For low resolution structures, fix the B values of Babinet's bulk solvent to
Maximum Likelihood Parameters
Monitoring
Run
Save or Restore
Close

Figure 16. Task window for running Refmac.

- ***Important!*** You should select the weighting term for the X-ray structure factors carefully. The default value of 0.3 is generally too high for typical data of 2.0-2.5 Å resolution, and will result in excessive distortion or mangling of the model. To increase the weight of geometric restraints, the X-ray weighting factor should be decreased. Typical values of the weighting factor are 0.05-0.20, depending on the resolution and quality of the data. You should adjust the weighting factor until the RMSD of bond lengths and bond angles are 0.010-0.020 Å and 1.5-2.0°, respectively. (You can determine the RMSD of bond lengths and angles by examining the REFMAC log file.) This degree of geometric constraint should generate acceptable structures that are appropriately dependent on the observed X-ray structure factors. Once you have determined an adequate X-ray weighting factor, you may use that value for the remainder of the refinement.
- Select *Babinet scaling*. This is typically more accurate than the default *simple scaling*, and results in more reasonable *b*-values for the protein structure. For low resolution data, it may be advantageous to fix the solvent *b*-value. Typical solvent *b*-values are between 100-400 $Å^2$, with 280 $Å^2$ being a commonly accepted optimal value.
- Start the job by selecting *Run…Run Now* at the lower left of the task window. The job will be entered into the job list in the CCP4i window, and you can monitor its status.
- When the job is finished, examine the log file from the *View Files from Jobs* menu in the administration functions pane of the CCP4i window to verify that the job has run correctly.

Visualizing Molecules and Electron Density Maps in O

O is a very powerful molecular visualization and model building program written by Alwyn Jones, Uppsala University, Sweden. Using O, it is possible to visualize the quality of fit of models to electron density data, and also to interactively alter the model to better fit the data. The latter of these activities, termed *rebuilding*, is essential to the refinement process. Although refinement programs are very sophisticated, it is not now possible for an automated refinement to find the best fit of model to data if the model is too far away from the correct solution. The purpose of rebuilding is to position the model in a more appropriate starting point for refinement to do its magic.

Starting and customizing O. The command for starting O is typically aliased to something easy to type. For example, `ono9` might be the alias used to run O version 9.0. Usually, the command is issued from the local directory from which you are working, so that you do not have to specify complete paths to your data files. O will ask for the locations of several files on startup, and normally the defaults should be accepted. If you want to read in a previously saved O session, provide the file name of a session file at the prompt to the first question. (*Note*: if running O in Windows, you should always read in the file `odat.odb` at the first prompt, to tell O where to find its internal data files. The file `odat.odb` is described later.)

O is rarely run without a great deal of personal customization, which is done by including a series of series of custom files in your working directory. Typical files include a personal

menu, special scripts to automate commonly used tasks, O-database files to modify how O works, and a list of commands to execute on startup. The last of these must be put into a file named `on_startup`. It is a good idea to give all O-database files the extension `.odb` so that you will know their intended use.

File 11
`menu_rowlett.odb`

```
.MENU                    T          41         24
colour_text red
STOP
colour_text white
Save_DB
colour_text magenta
Clear_flags
colour_text green
Yes
colour_text red
No
colour_text cyan
Centre_ID
Clear_ID
colour_text yellow
Dial_previous
Dial_next
colour_text cyan
Lego_CA
Lego_side_ch
Water_add
colour_text yellow
Grab_atom
Grab_fragment
Grab_residue
Move_zone
colour_text cyan
Flip_peptide
Refi_zone
Tor_residue
colour_text yellow
Dist_define
Neighbour_atom
Trig_reset
Trig_refresh
colour_text turquoise
@gen_symmetry
@redraw_solv
@redraw_map
@next_water
@skip_5_ca
@next_ca
```

You may create a custom menu by reading into O an appropriate O-database. File 11, `menu_rowlett.pdb` is an example of a handy O menu that keeps commonly-used commands close at hand and executable at the click of a mouse. Commands included in the personal menu will appear in a box on the O display screen. The position of this menu is normally controlled by the `on_startup` file. The first line in the file describes this as a menu file (`.MENU`) composed of text (`T`), 40 lines long (not counting the first line), with a maximum of 24 characters per line. The menu can be color-coded in blocks by adding `colour_text` lines; the other lines in the file are either existing O commands, or references to user script files; script files references can be recognized by the preceding `@` symbol.

Some useful script and database files are shown in Files 12-16. A brief description of the files is given in the comment section of the file, following the `!` symbols. File 17 is a useful O database file that modifies the way O labels residues picked by the mouse, displaying the residue type as well as the chain ID, *e.g.*, `Arg D160 CA` instead of `D160 CA`. File 18 is required when running O in the Windows operating system, as described previously, and must be read in before O starts.

Files 12-16

next_ca

```
! centers screen on next alpha-carbon and redraws
! electron density maps as defined in on_startup
centre_next atom_name = ca
fm_draw 2fofc
fm_draw fofc+
fm_draw fofc-
```

next_water

```
! centers screen on next solvent molecule and
! redraws electron density maps as define in on_startup
centre_next atom_name = o
fm_draw 2fofc
fm_draw fofc+
fm_draw fofc-
```

redraw_map

```
! redraws 2fofc and fofc+/- maps as
! defined in on_startup
fm_draw 2fofc
fm_draw fofc+
fm_draw fofc-
```

redraw_solv

```
! redraws solvent molecules and protein
! useful after using add_water command
! rename solv and hica as required
mol solv
zo ;end
mol hica
```

gen_symmetry

```
! generates nearby (<10 A) symmetry atoms
! rename molecule hica and alter radius as required
sym-sph hica sym 10.0
```

Files 17-18

resid.odb

```
.ID_TEMPLATE         T          2         40
%Restyp %RESNAM %ATMNAM
residue_2ry_struc
```

odat.odb

```
! edit deirectory to point to O data files
.odat t 1 50
C:/o/data/
```

The `on_startup` file, if present in the working directory, controls what commands and other functions O should perform each time the program is started. Normally, `on_startup` should not read in any `.pdb` files, but it is useful for it to read in and format electron density maps, menus, and any O databases desired. An example is shown in File 18, which reads in a custom menu, the file `resid.odb`, the $2F_o$-F_c map, and color codes both the positive and negative density in the F_o-F_c map according to its deviation in σ. It also positions all the menus on the screen so as not to interfere with the visualization of the molecule of interest.

File 18

`on_startup`

```
read menu_rowlett.odb
read resid.odb
win_open user_menu 0.9 1.0
win_open object_menu -1.20 0.7
win_open dial_menu -1.20 0.2
fm_file 2fofc.omap 2fofc C2
fm_file fofc.omap fofc+ C2
fm_file fofc.omap fofc- C2
Fm_setup 2fofc 20 ; 1 1 medium_blue
Fm_setup fofc+ 20 ; 3 5 white 4 cyan 3 blue
Fm_setup fofc- 20 ; 3 -3 red -4 orange -5 yellow
window_open density_1 -.55 -.9
window_open density_2 0.05 -.9
window_open density_3 0.65 -.9
```

Loading a molecule into O. A molecule can be loaded into O for inspection by issuing the command `pdb_read`. Commands can be entered in the graphics window or in the text window. Extended command sessions are best done in the text window. (*Note*: if running the Windows version of O, all commands must be entered into the text window.) When prompted, supply the filename to read in and a molecule name (6 letters or less) that you will use to identify the molecule in O. To make the molecule visible type `mol` *`molname`*, where *`molname`* is the name you supplied in `pdb_read`. To render the entire molecule, type `zone ;end`. If the molecule does not appear, it is probably not centered in your viewing area. Center the molecule by using a command such as `ce_atom a44`, which would center the screen on the α-carbon of residue 44 in chain A of the molecule. The graphical viewing enivronment of O is shown in Figure 17.

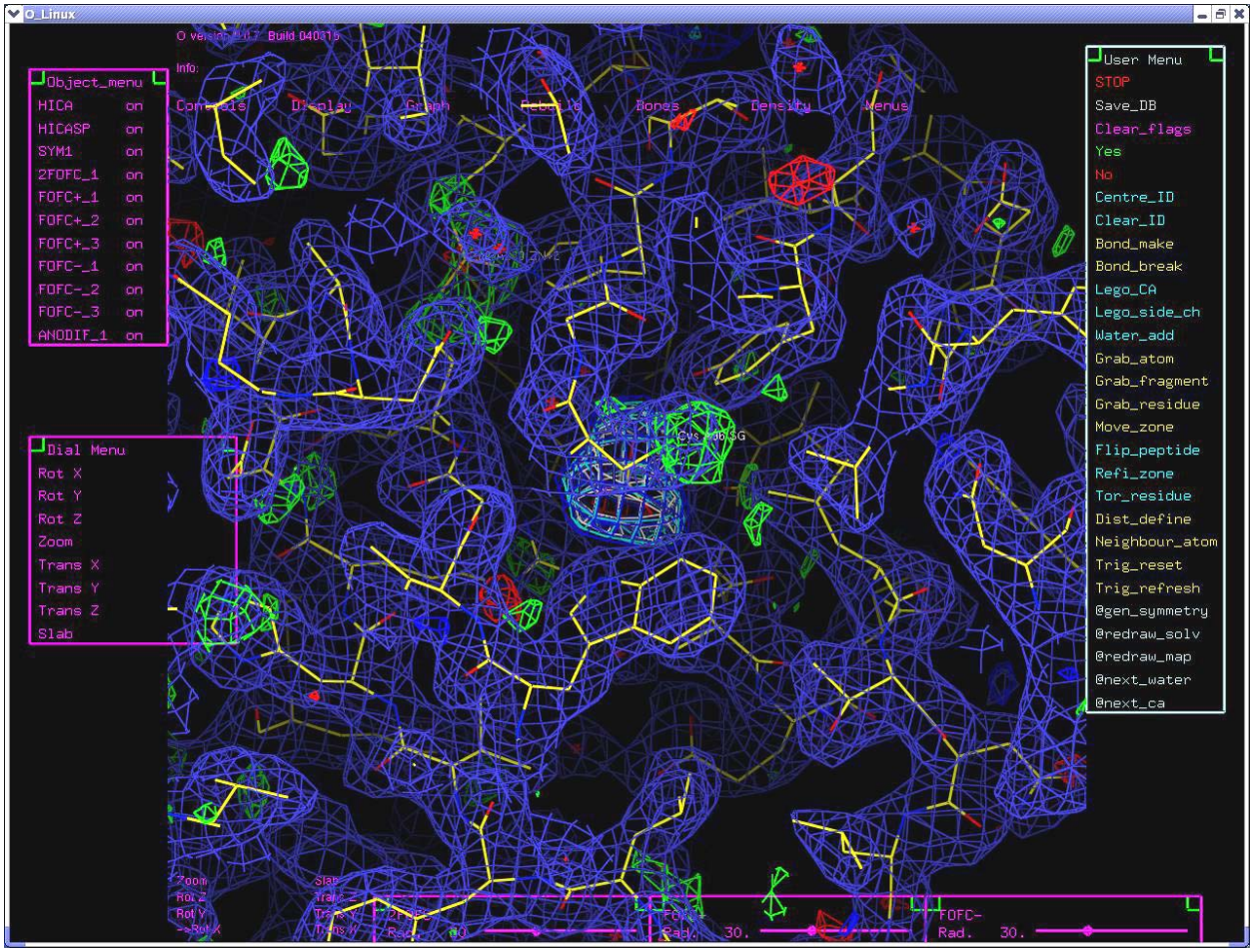


Figure 17. The O graphics window. The object and dial menus are on the left, the customizable user menu is on the right.

Inspecting a molecule in O. The molecule can be manipulated on the screen with the mouse. Press the right mouse button and drag to spin the molecule. To zoom, hold the middle mouse button while scrolling up and down. To slab (cut away) the molecule, hold the middle button while scrolling left and right. Pointing at an atom and clicking will display an identifying label. You may turn on or off the display of various objects in the model by clicking on the appropriate name in the object menu.

Generating symmetry atoms. It is frequently useful to generate symmetry-related atoms in the displayed model in order to observe interactions at protein-protein interfaces, or to get a more accurate view of an interfacial active site, etc. O must be initialized with the `sym_setup` command prior to generating symmetry atoms. The `sym_setup` command will prompt for the molecule name, unit cell dimensions and space group. If there is a CRYSTAL record in your input PDB file, the correct default values will be presented, otherwise you will have to enter them manually. `Sym_setup` need only be executed once. To generate symmetry related atoms around the currently selected atom, issue the command `sym_sphere` in the text or graphics window, or choose the command `@gen_symmetry` on the user menu installed by

`menu_rowlett.odb`. The default radius for plotting symmetry-related atoms is 10 Å, but this can be altered if desired.

Saving O sessions. Saving your work in progress is not only desirable, it's essential when working in O, as it is known to crash unpredictably. The entire state of your O project can be saved at any point by issuing the `save_db` command, or by clicking on `SAVE` in the custom menu. The first time you issue this command, you will have to supply a filename. O session files should be given the `.o` extension to help identify them. To retrieve and O session, simply read in the appropriate `.o` file at the initial prompt after starting O. The entire state of O at the time of the `save_db` command, including all `.odb` files loaded at that time, will be re-instated. `SAVE` your work frequently!

Basic Model Building Tasks in O

There are several common tasks in rebuilding models to better fit the electron density maps. Some of these are described here. Typically, after each refinement cycle, the model is inspected for conformity to the electron density, and modified as necessary to make it possible for the minimization algorithm to more easily find the best solution.

Mutating residues. One of the first tasks to complete when a structure is being solved by molecular replacement is to change the mismatched residues in the search model to conform with that of the target molecule. Before performing this task, the command `mut_setup` must be issued to initialize O for this task. Once `mut_setup` has been successfully executed, a residue is mutated by issuing the `mut_replace` command. You will be prompted separately for a molecule name and the name of a residue to change and what to change it to. If desired, you can do this all at once on the command line, *e.g.*, `mut_replace hica a181 phe` would change residue 181 in chain A of the molecule `hica` to a Phe residue. You will continue to be prompted for additional mutations until you type a blank return at the prompt. Once the `mut-replace` command has been successfully completed, the entire molecule will disappear from the screen. Redraw the molecule with the `zo ;end` command. The mutated residue(s) will display in purple. At this point it is useful to use the `lego_side_chain` or `tor_residue` commands to adjust the new side chain into its electron density. This task is described below.

Adjusting side chain conformation. Two approaches are available for manipulating side chain conformation. The `lego_side_chain` command allows the user to choose from among a population of commonly represented conformers of a particular side chain. Sometimes this is all that is necessary to achieve a fit close enough for minimization. For finer control of side-chain conformation the `tor_residue` command is preferred, as this will allow the adjustment of all the side-chain torsion angles as well as the phi and psi angles of the main chain.

Before using the `lego_side_chain` command it is necessary to issue the `lego_setup` command first to initialize O for this purpose. To choose from among common conformers of a side chain, issue the command `lego_side_chain`, or choose this command from the user menu, and click on the residue you wish to alter. Click in the fake dial box to update its menu, and scroll through the possible rotamers by holding down the left mouse button while scrolling over the dial box *Rotamer* entry. When you are satisfied with the selected

rotamer, click on *Yes* in the user menu. If you want to start over and discard changes, click on *No* in the user menu.

For finer control over side-chain conformation, issue the command `tor_residue` or choose this command from the user menu, and click on the residue you wish to alter. The various adjustable torsion angles will appear on the selected side chain along with their current values. (The values can be very useful when using this command to manually flip Asn, Gln, and His side chains by 180° to improve hydrogen bonding contacts.) Click in the fake dial box to update its menu, and several new items will appear corresponding to the various torsion angles in the residue. The various torsion angles can be changed by holding down the left mouse button while scrolling over the appropriate dial box entry. When you are satisfied with the selected changes, click on *Yes* in the user menu. If you want to start over and discard changes, click on *No* in the user menu.

Adjusting main chain conformation. It will often be necessary to re-orient the main chain as well as side chaings to better fit the electron density map. This is most conveniently done using the `grab_atom`, `grab_fragment`, and `grab_residue` commands. The `grab_atom` command will move individual atoms, `grab_fragment` will move sidechains or mainchain atoms as a group, and `grab_residue` will move the entire residue. Similar manipulations can be carried out with the `move_atom`, `move_fragment`, and `move_residue` commands, the main difference being that the `move` commands require the use of the dial box while the `grab` commands allow the use of the mouse to drag atoms, fragments, or residues about the screen. The grab commands will be described here.

To grab an item to move about the screen, issue the appropriate `grab` command or select it on the user menu, then click on the item to be moved while holding down the mouse button. The item can be moved in the x- and y-directions by simply moving the mouse. (*Note*: these movements are in 2-dimensional screen coordinates only, so it is usually wise to inspect the effects of your movement from several viewpoints to ensure that you have correctly placed the moved item in 3-dimensional space. You may continue to click and drag the item as many times as you wish. In addition, you may rotate the item about the initially selected atom on the x- or y-axes by holding down the CTRL key and the right mouse button. You may rotate the item about the initially selected atom on the z-axis byt holding down the CTRL key and the right and middle mouse buttons. When you are satisfied with the orientation of the moved item(s), select *Yes* from the menu else select *No* to cancel the operation. (*Note*: if you move more than one item, only the last item moved will be able to be canceled. It is recommended that you normally only move one item at a time, and complete that operation before proceeding. Alternatively, execute a SAVE prior to making complex `grab` operations so that it is possible to return to a known previous state.) Grab movements will probably seriously distort the normal bond angles and distances in the main and side chains, so the affected region of the model should be regularized before proceeding. After regularization, described below, it may be necessary to use the `grab` command(s) again to make minor adjustments. Usually 1-2 iterations of moving and regularization are sufficient to correctly place the model into the electron density.

Regularizing the model. The `refi_zone` command can be used to regularize a manually adjusted model to conform to normal, expected bond angles and disitances. To regularize a portion of the model `refi-zone` can be selected from the user menu, following by

clicking on two atoms in the model to define a range of atoms to be regularize. Select *Yes* on the user menu to accept the refinement, or *No* to cancel. Alternatively a command can be issued in the text or graphics window. For example, the command `refi_zone hica a44 a46` would regularize all atoms in chain A of the molecule `hica` for residues 44-46. Again, select *Yes* to accept and *No* to cancel the refinement.

Writing out a PDB file. After modifying a model it is likely that you will want to write out a new PDB file reflecting the changes you have made in the model. To write out a PDB file, issue the command `pdb_write` and specify a file name when prompted. Give the filename a `.pdb` extension to help identify it for later use.

Adding ligands and cofactors to a model in O

Many proteins contain non-protein cofactors such as metal ions and coenzymes. In addition, crystallized proteins may tightly associate with buffer molecules, inorganic ions, and precipitant molecules. It is often quite desirable to account for the electron density of these substances by adding them to the model.

Using HIC-UP. While it is possible to manually edit `.pdb` files to do this, normally it is much more convenient to download an existing, geometry-optimized model in `.pdb` format. A good source for such files is the HIC-UP server found at http://xray.bmc.uu.se/hicup/. This server contains over 4000 commonly encountered non-protein molecules that have been found to associate with proteins in the crystalline state. Normally one should download for each non-protein molecule a clean `.pdb` file, the CNS topology and parameter files (for guiding refinement in CNS), the O connectivity entry, and the O refi dictionary entry (if you intend on using `refi_zone` on the hetero molecule). The O connectivity entry should be added to a copy of `all.dat` and stored in your working directory. When O prompts for the location of `all.dat` during startup, point to the modified copy. The refi dictionary entry should be added to the `.bonds_angles` O datablock and saved in your local directory according to the file instructions. In all CNS refinement files, it will be necessary to add references to the topology and parameter files for each hetero molecule. Some ligands already handled by CNS, such as monocations, some monoanions, phosphate, and sulfate, need not have separate topology and parameter file references in CNS scripts.

Adding ligands to a protein. The simplest way to accomplish this is to use `pdb_read` to load the .pdb file of the desired ligand into an existing O session with the protein and electron density map displayed. Select and draw the molecule. Use `grab_residue` to drag the ligand to the appropriate location and fit it into its electron density. Once the ligand(s) is (are) in place, write out a `.pdb` file of the ligand(s) using the `pdb_write` command. Use MOLEMAN2 to edit the ligand `.pdb` file and give it a unique segment id. Then the protein `.pdb` file and the ligand `.pdb` can be combined using a text editor for further refinement or model-building.

Adding additional amino acids to the N- or C- terminus. You may discover during refinement that you can see additional electron density beyond the termini of your search model, which may not include the entire gene coding sequence. Additional amino acid residues can be added much as are ligands, except that excellent `.pdb` files for additional residues can be

obtained by extracting and copying appropriate portions of the protein molecule using a text editor. These protein fragments can be repositioned using the `grab_residue` command, and the "extra" amino acids can be written out to a separate `.pdb` file. To incorporate these extra amino acids into the model, renumber the residues appropriately in MOLEMAN2, and combine with the original protein `.pdb` file using a text editor. It might be useful to examine the combined model in O, and perform a `refi_zone` to clean up the geometry of the junction between the original protein and the extended residues.

Adding water molecules to a model using CNS and O

The first batch of water molecules can be added automatically by using the CNS script `waterpick.inp`. This will append coordinates to the end of your `.pdb` file that correspond to appropriate electron density found to match reasonable geometric constraints for hydrogen bonding to the protein molecule. The script includes instructions for naming water molecules (typically HOH) and attaching a segid (typically S). It is unlikely that `waterpick.inp` will place all the required water molecules in your structure, so it will be necessary to add the rest manually using O, as described below.

Start O and load your protein molecule and electron density maps as usual. Issue the `water_init` command. The command will prompt for a molecule name (typically SOLV), the number of water molecules to reserve space for (100-300 is enough, depending on the size of your protein molecule and the resolution of the data set), and the number of the first residue for the added water molecules. To make life easier later, ensure that this number is larger than the number of water molecules added so far. (You will usually add water molecules in several sittings, so you want to ensure you don't accidentally overwrite any existing water molecules.) To add a water molecule, center on an atom close to the electron density you would like to fill, and issue the command `water_add`. A new water molecule will appear as a little red star superimposed on top of the atom you previously centered on. Click on *Yes* to accept the existence of the new water molecule. Turn off the protein molecule (so you can see the water molecule more clearly) and move the water molecule into the electron density using `grab_residue`. When you are satisfied with the position of the water molecule—check it in 3 dimensions!—click on *Yes* to accept. Note: when you place a new water molecule, all others will be erased. Immediately after adding a water molecule redraw the solvent molecule using the menu item `@redraw_solv` found in the user menu loaded with `menu_rowlett.odb`. This will usually discourage you from filling up the same electron density with more than one water molecule. Navigate and repeat adding water molecules using `water_add` as required.

During subsequent rounds of refinement with added water molecules, you may find it necessary to add additional water molecules or remove them. Water molecules with poor occupancy and high *b*-factors typically have electron density that shrinks like a prune during repeated rounds of refinement. Water molecules with tiny electron density surrounding them and/or *b*-factors in excess of 50 are candidates for removal from the final model.

Visualizing Molecules and Electron Density Maps in Coot

Coot is a very powerful and easy-to-use molecular visualization and model building program written by Paul Emsley, University of York, England. Using Coot, it is possible to visualize the quality of fit of models to electron density data, and also to interactively alter the

model to better fit the data. The latter of these activities, termed *rebuilding*, is essential to the refinement process. Although refinement programs are very sophisticated, it is not now possible for an automated refinement to find the best fit of model to data if the model is too far away from the correct solution. The purpose of rebuilding is to position the model in a more appropriate starting point for refinement to do its magic.

Coot is specially designed to integrate well with the CCP4 suite of crystallography programs, so it is especially appropriate if you are using MOSFLM, SCALA, and Refmac. One of the many nice features of Coot is the ability to re-contour electron density maps on the fly using phase and intensity data written out by Refmac. It also has the ability to do very nice real-space refinements of segments of the model. Most new users of crystallography software will find Coot easier to use and to integrate into the model refinement environment than O.

Starting Coot and loading a molecule and electron density maps. The command for starting Coot is typically aliased to something like `coot`. Usually, the command is issued from the local directory from which you are working, so that you do not have to specify complete paths to your data files. When Coot is started, it always asks you if you want to run an auto-save file that stores the last saved state. Unless you want to start where you left off, you can click on *No*. To load a `pdb` file, select *File…Open Coordinates* and choose the appropriate file name. The selected molecule will be loaded and displayed as a stick model. To open a Refmac-generated MTZ file that contains phases and intensities for contourable maps, select *File…Auto Open MTZ* and choose the appropriate file name. Both $2F_o–F_c$ (blue) and $F_o–F_c$ (positive = green and negative = red) electron density maps will be automatically displayed. The graphical viewing environment of Coot is shown in Figure 18. By default, the $2F_o–F_c$ map is contoured at approximately 1.5σ and the $F_o–F_c$ is countoured at approximately 3.0σ. The countour settings can be changed by rolling the center wheel of the mouse. To select which map will be re-contoured by default, select *HID…ScrollWheel…Attach ScrollWheel to which map*.

Navigating and inspecting a molecule in Coot. The molecule can be manipulated on the screen with the mouse. Press the left mouse button and drag to spin the molecule. To zoom, press the right mouse button and drag. To slab (cut away) the molecule, press CTRL and the right mouse button and drag up and down. To navigate to a particular atom, select *Draw…Go To Atom* to open up the navigation window and select the desired protein chain, residue, and/or atom desired. To recenter on an atom visible on the screen, middle-click on it. To navigate to the next residue in the sequence, press the space bar.

Generating symmetry atoms and non-crystallographic symmetry traces. It is frequently useful to generate symmetry-related atoms in the displayed model in order to observe interactions at protein-protein interfaces, or to get a more accurate view of an interfacial active site, *etc.* To display symmetry atoms, choose *Draw…Cell & Symmetry* and tick *Yes* in the *Show Symmetry Atoms* box.

Many protein crystals display non-crystallographic symmetry, and this can often be used to advantage in the early rounds of refinement to increase signal to noise. Coot will automatically find non-crystallographic symmetry in your molecule and display overlay traces of symmetric protein chains upon request. To display non-crystallographic symmetry, choose

Edit…Bond Parameters and tick *Yes* in the *Draw Non-Crystallographic Ghosts* box. Non-crystallographic symmetry traces will be overlayed on the A chain of your protein.

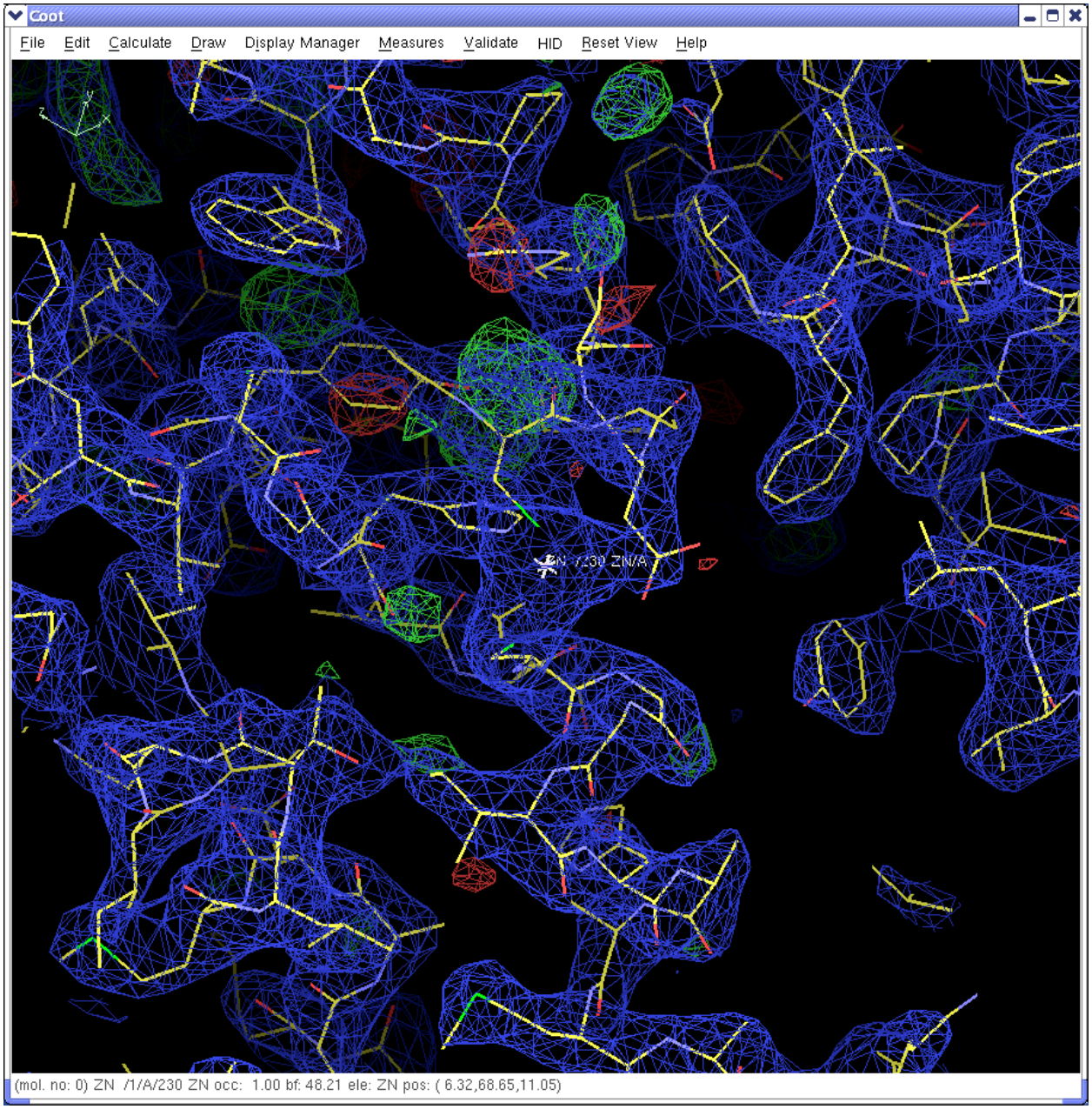

Coot
File Edit Calculate Draw Display Manager Measures Validate HID Reset View Help
ZN /230 ZN/A
(mol. no: 0) ZN /1/A/230 ZN occ: 1.00 bf: 48.21 ele: ZN pos: (6.32,68.65,11.05)

Figure 18. The Coot graphics window.

Recovering a session after a program crash. Coot is a work in progress, and has been known to crash unexpectedly. Fortunately, Coot is pretty good at saving your work as you go along, minimizing the chance that you will lose your work. To recover from a program crash up to but not including the last program edit, open Coot, read in the pdb you were last working on, and select *File…Recover Session* from the menu. After your PDB file has been updated, you may read in your electron density maps and resume.

Basic Model Building Tasks in Coot

Common tasks in rebuilding models to better fit the electron density maps are described here. Typically, after each refinement cycle, the model is inspected for conformity to the electron density, and modified as necessary to make it possible for the refinement program to more easily find the best solution.

Mutating residues. One of the first tasks to complete when a structure is being solved by molecular replacement is to change the mismatched residues in the search model to conform with that of the target molecule. In Coot, choose *Calculate…Mutate Residue Range*. In the dialog box choose the protein chain and the residue number or range to be mutated, and type in the one-letter amino acid code(s) for the mutation. If desired, you can autofit the mutated residue to the electron density map upon mutation by checking the appropriate box.

Ajdusting side chain conformation. Open the refinement task menu by selecting *Calculate…Model/Fit/Refine*. You now have several options to adjust side chain conformation on the task menu. *Auto Fit Rotamer* will select the best-fitting side-chain rotamer from a library of commonly observed conformations. This may be a good first attempt in some situations. You may also elect to interactively select a rotamer from the library by selecting *Rotamers*… from the task menu. To further refine this solution automatically, you can select *Real Space Refine Zone* and then click twice on any atom in the side chain (to define the side chain as the refinement zone). A dialog box will offer you the choice to accept or reject the fit, which will be highlighted in the graphics window.

For more precise control over side-chain fitting, *Edit Chi Angles* should be selected. Click on the side chain of the desired residue, and alter individual chi angles by sliding the mouse back and forth on the graphics screen. The new conformation will be highlighted. To quickly shift between chi angles, you can use the number keys: pressing *1* selects the first chi angle, *2* selects the second, *etc*. To complete the operation, select *Accept* or *Cancel* in the dialog box.

Adjusting main chain conformation. If you have to adjust the main chain trace, it is unlikely automated methods will work, else it would have been fixed already. Typically, the best way to adjust main chain conformation is by moving individual atoms and then regularizing the final result. To move atoms in the structure, select *Rotate/Translate Zone* from the Model/Fit/Refine task menu. Click on any atom in the residue you wish to move. To move an entire residue, simply click and drag. To move a single atom, -click and drag. Coot will automatically make and break atomic connections according to interatomic distance, so proceed cautiously to maintain the correct main- and side-chain connectivity! Select *OK* to accept changes, or *Cancel* to abandon.

Regularizing the model. It is often necessary to clean up manual adjustments by regularizing the structure so that it conforms to normal bond angles and lengths. This is especially helpful when adjusting the main chain. To do this, select *Regularize Zone* from the task menu. Click on two atoms in the structure between which the structure will be regularized, typically plus and minus at least one residue from the area in which manual changes were made so that changes can be blended into the overall main-chain trace.

Writing out a PDB file. To save your structural edits, select *File…Save Coordinates* from the main menu. You will be prompted for the molecule to save (several may be open in the graphics window at the same time) and be expected to select a filename. Give the filename a `.pdb` extension to help identify it for later use.

Adding Ligands and Cofactors using Coot

Adding ligands to a protein. Adding ligands and cofactors is ridiculously easy in Coot. From the main menu, select *File…Get Monomer* and enter the three letter code of the desired ligand, cofactor, or metal. A complete list of monomers can be found in the CCP4 documentation.[8] The selected molecule will be placed at the center of the display. Move the ligand to the desired location using *Rotate/Translate Zone* in the Model/Fit/Refine task menu. The coordinates for the cofactor can be written out as a separate PDB file for manual merging into the protein coordinate file, or the coordinate can be appended to the end of any displayed PDB file by selecting *Calculate…Merge Molecules*.

Adding additional amino acids to the structure. This is another easy Coot task. From the Model/Fit/Refine task menu, select *Add Terminal Residue…* and click on the terminus of the molecule you would like to add to. Coot will add an alanine residue and make its best guess of the appropriate conformation. You may have to mutate the added residue to the correct side chain and adjust its conformation to match the observed electron density.

Adding Water Molecules to a Model using CCP4 and Coot

Adding water molecules to a structure is most easily accomplished by running Refmac together with ARP_WATERS to automatically and iteratively place water molecules in the structure according to specified and unbiased electron density constraints. To enable ARP_WATERS, select the *Cycle with arp_waters…* option in Refmac (Figure 19). Under *Refinement Parameters*, select 10-20 cycles of ARP_WATERS. Careful selection of the *ARP_waters parameters* is necessary for successful automated placement of water molecules. In particular:

- The maximum number of new waters to be found per cycle should normally not exceed $0.08 \times N/R^3$, where N is the total number of protein atoms in the structure, and R is the resolution in angstroms. Therefore, for a 2.3 Å structure with 3000 protein atoms, the number of new solvent atoms found each cycle should not exceed 20.
- The threshold electron density for water addition is typically set to 3-4σ.
- The number of waters to be removed each cycle should normally be set to 25-100% of the number of water molecules to be found.
- The threshold electron density for water removal is typically set to 1σ.
- The water "chain" in the protein is typically labled W or S.
- The remaining CCP4i/ARP_WATERS defaults are acceptable.

[8] A description of CCP4-recognized monomers can be found at http://www.ccp4.ac.uk/html/lib_list.htm. For example, the zinc ion is ZN, bicarbonate ion is BCT, sulfate is SO4, phosphate is PO4, *etc.*

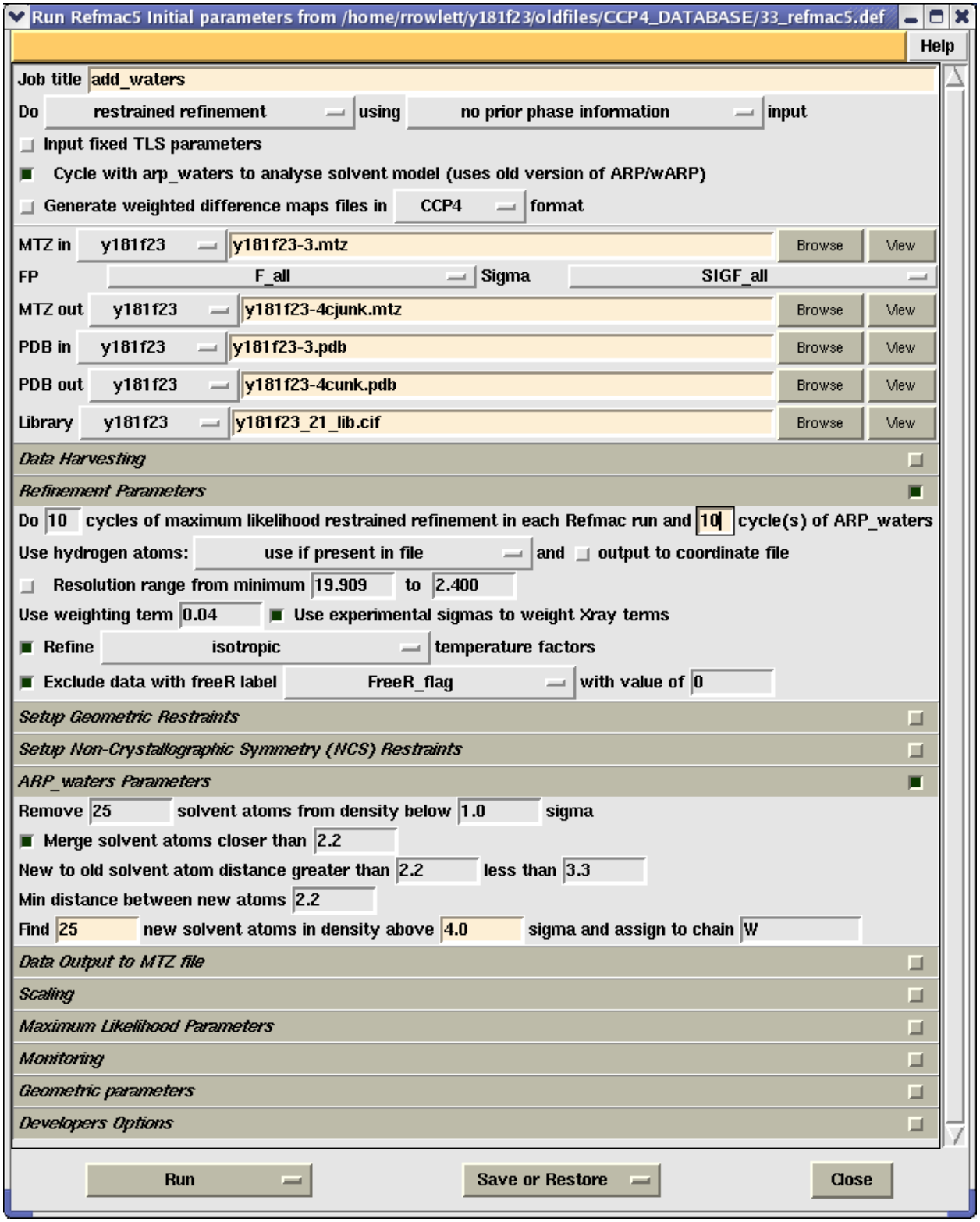

Run Refmac5 Initial parameters from /home/rrowlett/y181f23/oldfiles/CCP4_DATABASE/33_refmac5.def
Help
Job title add_waters
Do restrained refinement using no prior phase information input
Input fixed TLS parameters
Cycle with arp_waters to analyse solvent model (uses old version of ARP/wARP)
Generate weighted difference maps files in CCP4 format
MTZ in y181f23 y181f23-3.mtz Browse View
FP F_all Sigma SIGF_all
MTZ out y181f23 y181f23-4cjunk.mtz Browse View
PDB in y181f23 y181f23-3.pdb Browse View
PDB out y181f23 y181f23-4cunk.pdb Browse View
Library y181f23 y181f23_21_lib.cif Browse View
Data Harvesting
Refinement Parameters
Do 10 cycles of maximum likelihood restrained refinement in each Refmac run and 10 cycle(s) of ARP_waters
Use hydrogen atoms: use if present in file and output to coordinate file
Resolution range from minimum 19.909 to 2.400
Use weighting term 0.04 Use experimental sigmas to weight Xray terms
Refine isotropic temperature factors
Exclude data with freeR label FreeR_flag with value of 0
Setup Geometric Restraints
Setup Non-Crystallographic Symmetry (NCS) Restraints
ARP_waters Parameters
Remove 25 solvent atoms from density below 1.0 sigma
Merge solvent atoms closer than 2.2
New to old solvent atom distance greater than 2.2 less than 3.3
Min distance between new atoms 2.2
Find 25 new solvent atoms in density above 4.0 sigma and assign to chain W
Data Output to MTZ file
Scaling
Maximum Likelihood Parameters
Monitoring
Geometric parameters
Developers Options
Run
Save or Restore
Close

Figure 19. Refmac task window with ARP_WATERS option enabled.

Validating Structures

Before a structure is deposited with the Protein Data Bank, it is necessary to evaluate the proposed structure for its quality, including consistency with typical known bond lengths and angles, steric hindrance, and appropriate hydrogen bonding networks. Structures (`.pdb` files) can be uploaded and evaluated at the www.biotech.ebi.ac.uk:8400/ server. It is recommended that you perform a complete check. The following results should be especially scrutinized. Typically you should be most concerned with any check results labeled "bad".

- BPOCHK and BH2CHK—these examine polar residues for missing hydrogen bonds. Normally, all polar residues in proteins are fully engaged in hydrogen bonding to something. For each residue indicated to have missing hydrogen bonds, examine the structure carefully. Residues on the surface of the protein can usually be ignored, as they are likely hydrogen bonded to water that is not visible in the electron density maps. Modify the structure as required to correct missing hydrogen bonds, if possible.
- ANGCHK—any residues scoring>0.5 should be investigated, as high scores indicate residues found in unusual conformers. If the electron density clearly justifies the observed conformation, no changes are necessary. Otherwise, modify the structure as required to bring the residue into conformity.
- HNQCHK—this utility checks for Asn, Gln, or His residues that would establish better hydrogen bonding interactions if flipped 180°. Use `tor_residue` in O to flip the required side chains 180°.
- Ramachandran plot—examine the Ramachandran plot to determine if most residues (except for mostly Gly) are in the preferred ϕ and ψ angle regions. Typically at least 90% of the residues should be in the preferred regions. Investigate any residues other than Gly that are in non-preferred conformations, and make corrections if necessary.
- WHATIF—investigate all "bad" results in detail. In particular, examine "bumps" (steric crowding) to see if they are real or simply the result of large *b*-factors.

Once structural problems have been resolved, re-refine the structure using only `bgroup.inp` and `bindividual.inp` to re-calculate proper *b*-factors (CNS) or run one cycle of Refmac without ARP_WATERS. Then repeat structural validation. When all fixable structural anomalies have been addressed, you are ready to submit the structure to the Protein Data Bank at http://www.rcsb.org.

Additional Resources

- CCP4: http://www.ccp4.ac.uk/
- Crystallography and NMR System: http://cns.csb.yale.edu/v1.1/
- Coot: http://www.ysbl.york.ac.uk/~emsley/coot/doc/user-manual.html
- The O files: http://www.imsb.au.dk/~mok/o/
- O for Morons: http://seqaxp.bio.caltech.edu/www/hhmi_manuals/morons/o_for_morons.html
- Uppsala Software Factory: http://xray.bmc.uu.se/~gerard/manuals/
- A to Z of O: http://xray.bmc.uu.se/alwyn/A-Z_of_O/A-Z_frameset.html